Fine-Tuned Convex Approximations of Probabilistic Reachable Sets under Data-driven Uncertainties

Pengcheng Wu, Sonia Martinez, *Fellow, IEEE*, Jun Chen, *Member, IEEE*

***Abstract*—This paper proposes a mechanism to fine-tune convex approximations of probabilistic reachable sets (PRS) of uncertain dynamic systems. We consider the case of unbounded uncertainties, for which it may be impossible to find a bounded reachable set of the system. Instead, we turn to find a PRS that bounds system states with high confidence. Our data-driven approach builds on a kernel density estimator (KDE) accelerated by a fast Fourier transform (FFT), which is customized to model the uncertainties and obtain the PRS efficiently. However, the non-convex shape of the PRS can make it impractical for subsequent optimal designs. Motivated by this, we formulate a mixed integer nonlinear programming (MINLP) problem whose solution result is an optimal n sided convex polygon that approximates the PRS. Leveraging this formulation, we propose a heuristic algorithm to find this convex set efficiently while ensuring accuracy. The algorithm is tested on comprehensive case studies that demonstrate its near-optimality, accuracy, efficiency, and robustness. The benefits of this work pave the way for promising applications to safety-critical, real-time motion planning of uncertain dynamic systems.**

***Note to Practitioners*—This study is motivated by the realization of safety-critical real-time motion planning for a dynamic system under uncertainties. A popular method used to guarantee the safe operation of uncertain dynamic systems is reachability analysis. However, this method may not work well in the face of the following challenges: unbounded uncertainties, unknown distributions, generality, convexity, and efficiency. To address these issues, we first present a data-driven approach to model arbitrary unknown uncertainties and obtain a set encompassing the system states with high confidence; Then we propose an algorithm to efficiently find a tight convex polygon approximation for the set. This clearly benefits real motion planning. When considering collision avoidance in motion planning, a tight convex approximation allows a larger feasible search area which may provide a better-planned trajectory. Also, the efficiency of the algorithm ensures that the motion planning can be realized in real-time.**

***Index Terms*—Probabilistic Reachable Set, Convex Approximation, Uncertain Dynamic System, Kernel Density Estimator, Mixed Integer Nonlinear Programming**

I. Introduction

The deployment of autonomous systems in urban ground and air traffic environments requires the development of new real-time collision detection and resolution algorithms [1]–[3]. However, widespread uncertainties in such systems raise important concerns about their safety [4]. Such uncertainties may have various causes, such as epistemic, given a lack of understanding of the underlying mechanism, the gap between simple assumptions and complex reality, or aleatoric, given the inherent randomness in the systems, or the errors of sensor measurements [5]–[8]. A popular method that is used to guarantee the safe operation of uncertain dynamic systems is based on reachability analysis [9]. This set-based method computes the reachable set of states that are accessible by a stochastic dynamic system from all initial conditions and all admissible inputs and parameters [10]. Reachability analysis has increasingly attracted significant attention from researchers in the past decades [11]. In prior literature, accounting for the knowledge of uncertainties, many algorithms have been proposed to utilize detailed system information to figure out a reachable set. However, existing algorithms may not work well or even fail in the face of the following challenges:

1) Unboundedness: When uncertainties are unbounded, it may be impossible to figure out a bounded reachable set for an uncertain dynamic system. However, most real-world applications need a bounded reachable set due to physical limits.

2) Unknown Distributions: Uncertainties are often assumed to obey Gaussian distributions in most literature. However, in practice, the uncertainties may be non-Gaussian or even empirical due to unknown system interconnections and nonlinearity [12], [13].

3) Generality: To compute reachable sets, many algorithms require detailed system information given *a priori*. However, in some applications like complex cyber-physical systems that are only accessible through experiments or simulations, this detailed information is unavailable. Many algorithms also rely on strong assumptions about the dynamics or uncertainties, which are often unrealistic.

4) Convexity: The reachable set computed by most algorithms is irregular or non-convex, which is hard to handle when it comes to the resolution of collision avoidance in practice.

5) Efficiency: In order to realize real-time motion planning and control of uncertain dynamic systems, the reachable set must be computed efficiently. However, many algorithms are computationally expensive.

To address the challenges, in what follows, we will first review some related literature, and then introduce the contributions of our paper.

Pengcheng Wu is with the Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093, and also with the Department of Aerospace Engineering, San Diego State University, San Diego, CA 92182 pcwupat@ucsd.edu, pwu@sdsu.edu

Sonia Martinez is with the Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093 soniamd@eng.ucsd.edu

Jun Chen is with the Department of Aerospace Engineering, San Diego State University, San Diego, CA 92182 jun.chen@sdsu.edu

A. Literature Review

A classical way to compute reachable sets for bounded uncertainties is given by performing a Hamilton-Jacobi (HJ) reachability analysis. It formulates the problem as a partial differential equation which can be solved by numerical methods. While computationally tractable in lower dimensions, the method scales poorly to higher dimensions [1], [14]. Also, HJ reachability analysis requires detailed information on dynamics or uncertainties and relies on strong assumptions like there are no time correlations of parameter uncertainties, which makes it unavailable in some realistic scenarios [1].

When uncertainties are unbounded, it is often impossible to guarantee collision avoidance with 100% certainty as the reachable set becomes unbounded. Instead, researchers seek to identify a trajectory with a probability of collision bounded by a certain risk bound. One direct way to realize this goal is to evaluate the probability of collision of a generated candidate trajectory. The candidate trajectory will not be adopted until the probability of collision is below the risk bound. However, there is no closed-form expression to evaluate the probability of collision, which poses difficulties to this approach [15], [16]. Another indirect but effective method consists of computing a probabilistic reachable set (PRS) instead of an exact one [17], [18]. A PRS is a set of possible states that a system can reach with a certain probability. By means of this, the stochastic constraint on the system, which sets a limit on the collision probability, can be converted to a deterministic constraint by which the generated trajectory does not intersect with the PRS [19]. An advantage of this method is that it is not only applicable for unbounded uncertainties but also for bounded ones. As the risk bound approaches 0%, the PRS becomes a superset of the reachable set [17]. In this paper, we adopt the PRS approach.

Many existing works assume a Gaussian distribution of the uncertainty [20], [21]. Blackmore et al. [19] use convex polygons (or polytopes) determined by the mean and covariance of the Gaussian distribution as a proxy of PRS. Wu and Chen et al. [4], [6] directly use the level sets of the Gaussian distribution to serve as PRS. However, in many scenarios uncertainties are of non-Gaussian type [12]. Even in that case, some researchers still make use of Gaussian distributions to approximate non-Gaussian ones [22]. Unfortunately, this approach can not guarantee that the risk of a generated trajectory is still within a given bound [13]. Instead, Aoude et al. [23] apply a particle method to evaluate the probability of collision in the process of motion planning. Calafiore and Campi [24] use a scenario approach by which stochastic constraints are sampled to obtain a convex optimization problem whose solution is feasible for the original stochastic constraints with high confidence. However, all of the above typically require a large number of samples for probabilistic assurance, which is neither feasible nor computationally efficient. Alternatively, Han and Jasour et al. [13] employ a moment-based method to characterize non-Gaussian PRS. However, moment information is assumed to be known *a priori* in their works, which is not realistic as this information can only be learned online from sensor measurements [25]. Moreover, the PRS is often irregular, which is challenging to handle in practice.

Such drawbacks motivate the study of data-driven reachability analysis, using data obtained from experiments or simulations [25]–[27]. Devonport et al. [11] use the level sets of a Christoffel function to estimate the reachable sets. The Christoffel function is obtained from the samples of the arbitrary unknown uncertainties, without requiring any detailed information *a priori*. However, the level sets estimating the reachable sets may be irregular or non-convex, which is hard to deal with in practice. Motivated by this, Lew et al. [1] employ a conservative convex hull of data samples for reachable set approximation. The number of sides of the convex hull is unspecified, leading to an arbitrary number of constraints in subsequent design problems. A bounding box can enclose data samples with a fixed number of sides but is more conservative [28]. All the works above are intended to estimate reachable sets given bounded uncertainties and are not accurate approximations of the reachable sets because lacking uncertainty quantification. For arbitrary probability distributions, including unbounded ones, a natural way of approximating reachable sets is via empirical or data-fitted probability density functions (PDF), whose level sets serve as the reachable sets. In this paper, we employ this approach considering kernel density estimators (KDE) [29]. Our goal is to address important limitations of adopting such a method such as the lack of closed-form expressions for KDEs, the irregular shapes of their level sets, and the conservativeness and inaccuracy of taking convex hulls or bounding boxes of the level sets. Some researchers have considered dynamic systems with time delays [30], which is not done in this manuscript. The extension of the proposed methodology to scenarios where time delay matters is left for future work.

B. Contributions and Organization

This paper proposes an efficient convex approximation of PRS of uncertain dynamic systems while ensuring their accuracy. Our approach consists of two parts: (i) A data-driven uncertainty quantification; (ii) An optimization problem formulation to find the convex approximation.

The major contributions of this paper are the following:

1) We present an algorithm that efficiently finds the PRS via KDE accelerated by a fast Fourier transform (FFT). FFT-based KDE is customized to estimate the PDF under arbitrary unknown uncertainties and therefore its level sets can serve as a proxy for the PRS of the uncertainties. As an efficient data-driven approach, FFT-based KDE can learn from data online, and thus it neither requires *a priori* information nor relies on strong assumptions;

2) We formulate an optimization problem of mixed integer nonlinear programming (MINLP), resulting in an optimal n sided convex polygon that approximates the PRS. As opposed to a convex hull, users can arbitrarily customize the number of sides. The optimization problem is established not by using data samples directly but by weighted grid points from KDE, which makes it tractable. Compared with bounding boxes, the result obtained by this approach is more tight and accurate;

3) We develop a heuristic algorithm to efficiently solve the formulated MINLP problem. This algorithm performs

weighted sampling to select the representative grid points from all the grid points from KDE, which significantly reduces computational complexity. Comprehensive case studies demonstrate that this algorithm enjoys the benefits of accuracy, efficiency, near-optimality, and robustness while providing a convex approximation for the PRS.

The rest of this paper is organized as follows. In Section II, we formally define the concept of PRS and state the problem as providing a convex approximation for it. In Section III, we use KDE accelerated by FFT to model arbitrary probability distributions and present an algorithm to find the PRS. In Section IV, we formulate an MINLP optimization framework, the solution of which leads to a convex polygon approximation of the PRS. In Section V, we develop a heuristic algorithm to efficiently solve the formulated MINLP. In Section VI, we conduct comprehensive case studies to demonstrate the performance of our proposed algorithm. In Section VII, we draw conclusions and make suggestions for future research. In addition, rigorous proofs of some propositions in this paper are given in Section VIII Appendix.

II. Problem Statement

In this section, we take into account an uncertain discrete-time dynamic system and introduce the concept of reachable set and PRS of the system. Then, we present the goal of this paper.

Consider a general discrete-time dynamic system of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k, \boldsymbol{w}_k\right), \tag{1}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the system state, $\boldsymbol{u}_k \in \mathcal{U}_k$ is an uncertain control input, $\boldsymbol{\theta}_k \in \Theta$ is an uncertain parameter, $\boldsymbol{w}_k \in \mathbb{W}$ is an uncertain disturbance, and $\boldsymbol{x}_0 \in \mathcal{X}_0$ is an initial state. Unlike most prior works [1], in this paper, the sets $\mathcal{U}_k \subset \mathbb{R}^m, \Theta \subset \mathbb{R}^p, \mathbb{W} \subset \mathbb{R}^q$, and $\mathcal{X}_0 \subset \mathbb{R}^n$ can be unbounded. Given the uncertainties of $\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k, \boldsymbol{w}_k$, then $\boldsymbol{x}_{k+1}$ is a random vector which obeys an unknown probability distribution.

Formally, the concept of reachable set is introduced as follows.

Definition 1 (Reachable Set [31])**.** At time T, the *reachable set* $\mathcal{X}_T$ of the dynamic system in Eq. (1) is defined to be

$$\begin{aligned}\mathcal{X}_T := \big\{\boldsymbol{x}_T \in \mathbb{R}^n : \boldsymbol{x}_k = \boldsymbol{f}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{\theta}_{k-1}, \boldsymbol{w}_{k-1}\right), \\ \boldsymbol{x}_0 \in \mathcal{X}_0, \boldsymbol{u}_{k-1} \in \mathcal{U}_{k-1}, \boldsymbol{\theta}_{k-1} \in \Theta, \boldsymbol{w}_{k-1} \in \mathbb{W}, \\ k \in \{1, \ldots, T\}\big\},\end{aligned}$$

where $\boldsymbol{x}_0$ is the initial state and $\mathcal{X}_0$ is the initial set.

Observe that, since the sets $\mathcal{U}_{k-1}, \Theta, \mathbb{W}$, and $\mathcal{X}_0$ are not limited to bounded sets, the reachable set $\mathcal{X}_T$ may be unbounded. Hence, it may be infeasible to find a bounded set in which the state is guaranteed to lie. Instead, we expect to find a bounded set such that the probability of the state lying in the bounded set is greater than a confidence level. To this end, we formally introduce the concept of the PRS as follows.

Definition 2 (Probabilistic Reachable Set (PRS) [18])**.** At time k, a bounded set $\tilde{\mathcal{X}}_k$ is defined to be a *probabilistic reachable set* (PRS) of the dynamic system in Eq. (1) at confidence level α if and only if

$$\Pr(\boldsymbol{x}_k \in \tilde{\mathcal{X}}_k) \geq \alpha.$$

A PRS is a set of possible states that a system can reach with a certain probability. Note that $\tilde{\mathcal{X}}_k$ may not exist. When $\alpha = 100\%$, if there exists a PRS $\tilde{\mathcal{X}}_k$, then $\tilde{\mathcal{X}}_k$ is a superset of the reachable set.

In this paper, we assume the dynamic system in Eq. (1) is two-dimensional, and leave the general dimension case for future work. The goal of this paper is twofold: 1) We aim to put forward a data-driven approach to model the arbitrary unknown uncertainties and obtain a PRS $\tilde{\mathcal{X}}_k$ of the dynamic system; 2) Motivated by the realization of safety-critical real-time motion planning for uncertain systems, we aim to approximate the PRS $\tilde{\mathcal{X}}_k$ by means of a bounded set that satisfies: i) **Convexity**: The boundary of the set is an n sided convex polygon; ii) **Efficiency**: The computation of this set is tractable; iii) **Accuracy**: The probability of the state $\boldsymbol{x}_k$ lying in the set is close to the prescribed confidence level α; iv) **Optimality**: The area of the set is as small as possible (not too conservative) while ensuring accuracy.

III. PRS Identification

In this section, we model an arbitrary unknown probability distribution through FFT-based KDE, and develop an online algorithm to find a PRS at confidence level α.

A. FFT-Based KDE

We first briefly review FFT-based KDE. The concept of KDE is introduced below.

Definition 3 (Kernel Density Estimator (KDE) [29])**.** Let $\boldsymbol{x}_k \in \mathbb{R}^2, k \in \{1, \ldots, M\}$ be M data samples drawn from a 2-variate probability distribution given by a PDF $f : \mathbb{R}^2 \mapsto \mathbb{R}$. The *kernel density estimator* (KDE) $\hat{f} : \mathbb{R}^2 \mapsto \mathbb{R}$ to approximate the PDF f is defined to be

$$\hat{f}(\boldsymbol{x}) = \frac{1}{M}\sum_{k=1}^{M} K(\boldsymbol{x} - \boldsymbol{x}_k),$$

where the function $K : \mathbb{R}^2 \mapsto \mathbb{R}$ is the *kernel function* which is a symmetric 2-variate density function.

The choice of the kernel function K is not critical to the accuracy of KDE, so we use the standard 2-variate Gaussian kernel in this paper

$$K(\boldsymbol{x}) = (2\pi)^{-1}\det(\mathbf{H})^{-\frac{1}{2}}\exp(-\frac{1}{2}\boldsymbol{x}^\top\mathbf{H}^{-1}\boldsymbol{x}), \tag{2}$$

where $\mathbf{H}$ is a symmetric and positive definite bandwidth matrix. The choice of bandwidth matrix is crucial to the accuracy of KDE approximating PDF. In this paper, we choose the bandwidth matrix using Silverman's rule [32].

Consider a mesh of N^2 *grid points* $\{(x_i, y_j) \in \mathbb{R}^2 : i, j \in \{1, \ldots, N\}\}$ equally spaced on the plane $\mathbb{R}^2$, the boundary of which encompasses all M data samples $\boldsymbol{x}_k \in \mathbb{R}^2, k \in \{1, \ldots, M\}$. The mesh of grid points can be viewed as a map

$\boldsymbol{g} : \{(i,j) : i,j \in \{1,\dots,N\}\} \mapsto \{(x_i, y_j) \in \mathbb{R}^2 : i,j \in \{1,\dots,N\}\}$ given by

$$(x_i, y_j) = \boldsymbol{g}(i,j),\ i,j \in \{1,\dots,N\}, \tag{3}$$

where (i,j) is the index of a grid point whose coordinate is (x_i, y_j). Throughout the rest of this paper, a coordinate $\boldsymbol{g}(i,j)$ is abbreviated as $\boldsymbol{g}_{ij}$.

The KDE $\hat{f}$ given in Definition 3 can be evaluated on the mesh of N^2 grid points $\boldsymbol{g}$ in Eq. (3), which yields

$$\hat{f}_{ij} := \hat{f}(\boldsymbol{g}_{ij}) = \frac{1}{M}\sum_{k=1}^{M} K(\boldsymbol{g}_{ij} - \boldsymbol{x}_k),\ i,j \in \{1,\dots,N\}.$$

One way to significantly accelerate the evaluation of KDE is *linear binning*. Instead of placing the kernel functions K on M data samples, we can place them on N^2 grid points $\boldsymbol{g}$ weighted by grid counts $c_{uv}, u, v \in \{1,\dots,N\}$. A grid count c_{uv} is a weight chosen to represent the amount of data samples near a grid point $\boldsymbol{g}_{uv}$. To obtain c_{uv}, we can go through every data sample and assign weights to its neighbouring grid points, as shown in Fig. 1.

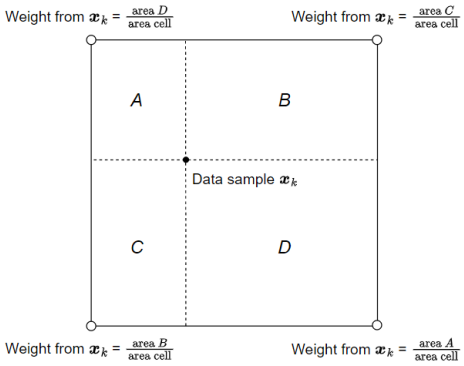


Fig. 1. Graphical representation of 2-variate linear binning

Using the strategy of linear binning, the *binned* KDE $\tilde{f} : \mathbb{R}^2 \mapsto \mathbb{R}$ to approximate the KDE $\hat{f}$ in Definition 3 is given by

$$\tilde{f}(\boldsymbol{x}) = \frac{1}{M}\sum_{u=1}^{N}\sum_{v=1}^{N} K(\boldsymbol{x} - \boldsymbol{g}_{uv})c_{uv}, \tag{4}$$

where c_{uv} is the grid count of the grid point $\boldsymbol{g}_{uv}$ indexed by (u,v).

Evaluating the binned KDE $\tilde{f}$ in Eq. (4) on the mesh of N^2 grid points $\boldsymbol{g}$ yields

$$\tilde{f}_{ij} := \tilde{f}(\boldsymbol{g}_{ij}) = \frac{1}{M}\sum_{u=1}^{N}\sum_{v=1}^{N} K(\boldsymbol{g}_{ij} - \boldsymbol{g}_{uv})c_{uv}, \quad i,j \in \{1,\dots,N\}, \tag{5}$$

which shows that the matrix $\tilde{\boldsymbol{f}} := [\tilde{f}_{ij}]$ is the discrete convolution of $\boldsymbol{c} := [c_{uv}]$ and $\boldsymbol{k} := [K(\boldsymbol{g}_{uv})]$.

The discrete convolution can be computed by FFT, which significantly saves computational time. Let $\boldsymbol{C}$ and $\boldsymbol{K}$ be FFT of $\boldsymbol{c}$ and $\boldsymbol{k}$, and $\tilde{\boldsymbol{F}}$ be the element-wise product of $\boldsymbol{C}$ and $\boldsymbol{K}$. Then $\tilde{\boldsymbol{f}}$ can be extracted from the inverse FFT of $\tilde{\boldsymbol{F}}$. By doing so, we can obtain the binned KDE $\tilde{f}_{ij},\ i,j \in \{1,\dots,N\}$ in Eq. (5) to approximate the PDF f of an arbitrary unknown probability distribution in real-time [15], [33].

B. Online Computation of PRS

For real-time motion planning of uncertain dynamic systems, we propose an efficient data-driven algorithm using FFT-based KDE to capture PRS of an unknown static uncertainty whose data samples are gradually observed. See **Algorithm 1**.

In the pseudo-code of Algorithm 1, the symbol ϵ represents the tolerance of confidence error. From Line 10 through 17, FFT is employed to accelerate the computation of the discrete convolution of $\boldsymbol{c}$ and $\boldsymbol{k}$ to obtain the binned KDE $\tilde{\boldsymbol{f}}$ evaluated on the mesh of N^2 grid points $\boldsymbol{g}$. From Line 18 through 33, a bisection search is implemented to find the critical binned KDE value C^{kde} at which the confidence level α is achieved. The level set of the KDE corresponding to C^{kde} is the PRS that we aim to find.

As the number of the collected data samples increases, KDE captured from those data samples will be convergent to the true PDF of the unknown uncertainties. However, considering more data samples requires a longer evaluation time. Fig. 2 shows how increasing data samples influence the evaluation time of identifying a PRS at confidence level 95% on a mesh of 128^2 grid points through implementing Algorithm 1. As the number of data samples increases, the proposed Algorithm 1 can rapidly recompute. For example, when the number of data samples is within the range of 10^3 to 10^4, the evaluation time of implementing Algorithm 1 is always under $0.01\,\text{s}$. This result demonstrates the computational power of our proposed Algorithm 1 and thus it can handle streaming data in a reasonable way. However, it cannot guarantee to deal with uncertainties that can evolve with time, and this will be part of our future work.

Moreover, note that the shape of a PRS obtained in this way is often irregular, which precludes its use in subsequent applications requiring an optimal design. This will be solved in the next section.

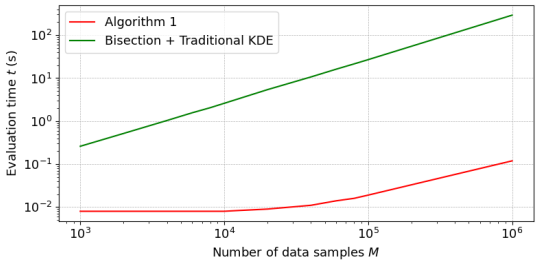


Fig. 2. Evaluation time of Algorithm 1 with respect to the increasing number of data samples

IV. MINLP Formulation for PRS Approximation

In this section, we formulate the convex approximation problem stated in Section II as an MINLP optimization problem for the PRS of a two-dimensional system subject to arbitrary uncertainties.

Algorithm 1 PRS Identification Algorithm

function GENDS(M)
  Generate M data samples $\boldsymbol{x}_k, k \in \{1, \ldots, M\}$
  return $\boldsymbol{x}_k$

function MESHGRID($\boldsymbol{x}_k$, N)
  Get $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ from data samples $\boldsymbol{x}_k$
  $\boldsymbol{h} = \text{linspace}(x_{\min}, x_{\max}, N)$
  $\boldsymbol{v} = \text{linspace}(y_{\min}, y_{\max}, N)$
  $\boldsymbol{g} =$ Cartesian product of $\boldsymbol{h}$ and $\boldsymbol{v}$
  return $\boldsymbol{g}$

function FFTKDE($\boldsymbol{x}_k$, $\boldsymbol{g}$)
  Obtain grid counts $\boldsymbol{c}$ for all grid points $\boldsymbol{g}$ using $\boldsymbol{x}_k$
  Evaluate kernel functions $\boldsymbol{k}$ on all grid points $\boldsymbol{g}$
  $\boldsymbol{C} = \text{FFT}(\boldsymbol{c})$
  $\boldsymbol{K} = \text{FFT}(\boldsymbol{k})$
  $\boldsymbol{F} =$ the element-wise product of $\boldsymbol{C}$ and $\boldsymbol{K}$
  $\tilde{\boldsymbol{f}} = \text{iFFT}(\boldsymbol{F})$
  return $\tilde{\boldsymbol{f}}$

function BISECSEARCH($\tilde{\boldsymbol{f}}$, α, ϵ)
  $low = \min(\tilde{\boldsymbol{f}})$
  $up = \max(\tilde{\boldsymbol{f}})$
  while $low < up$ **do**
    $mid = (low + up)/2$
    $\boldsymbol{z}^{\text{bin}} = (\tilde{\boldsymbol{f}} \geq mid) \cdot 1$
    $\boldsymbol{z}^{\text{mix}} =$ element-wise product of $\boldsymbol{z}^{\text{bin}}$ and $\tilde{\boldsymbol{f}}$
    $pr = \text{sum}(\boldsymbol{z}^{\text{mix}})/\text{sum}(\tilde{\boldsymbol{f}})$
    if $\text{abs}(pr - \alpha) \leq \epsilon$ **then**
      $C^{\text{kde}} = mid$
      return $\boldsymbol{z}^{\text{bin}}$, C^{kde}
    else if $pr < \alpha$ **then**
      $up = mid$
    else
      $low = mid$
  return $Failure$

function FINDPRS(M, N, α, ϵ)
  $\boldsymbol{x}_k =$ GENDS(M)
  $\boldsymbol{g} =$ MESHGRID($\boldsymbol{x}_k$, N)
  $\tilde{\boldsymbol{f}} =$ FFTKDE($\boldsymbol{x}_k$, $\boldsymbol{g}$)
  $\boldsymbol{z}^{\text{bin}}$, $C^{\text{kde}} =$ BISECSEARCH($\tilde{\boldsymbol{f}}$, α, ϵ)
  return $\boldsymbol{g}$, $\tilde{\boldsymbol{f}}$, $\boldsymbol{z}^{\text{bin}}$, C^{kde}

FINDPRS(M, N, α, ϵ)

Recall that, after evaluating Eq. (5) through the implementation of Algorithm 1, we have obtained that each grid point $\boldsymbol{g}_{ij},\ i,j \in \{1,\ldots,N\}$ has a binned KDE value $\tilde{f}_{ij},\ i,j \in \{1,\ldots,N\}$, respectively. We define a normalized weight matrix $\boldsymbol{w} := [w_{ij}]$ whose element is

$$w_{ij} := \frac{\tilde{f}_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N}\tilde{f}_{ij}}, \quad i,j \in \{1,\ldots,N\}. \tag{6}$$

The MINLP will be established using these weights, and not with the original data samples. This makes it more tractable because the number of grid points is far less than the number of data samples, significantly reducing the number of decision variables and constraints.

A. Problem Description and Objective Function Formulation

The optimization goal is to find a minimum n sided convex polygon to approximate the PRS efficiently and accurately, by determining which weighted grid points should lie inside the polygon.

To this end, we introduce three types of decision variables to formulate the MINLP optimization problem:

- $2n$ continuous variables $a_k, b_k \in \mathbb{R},\ k \in \{1,\ldots,n\}$;
- $N^2 n$ binary variables $l_{ij}^k \in \{0,1\},\ i,j \in \{1,\ldots,N\},\ k \in \{1,\ldots,n\}$;
- N^2 binary variables $z_{ij} \in \{0,1\},\ i,j \in \{1,\ldots,N\}$.

Here the integer $n \geq 3$ is the number of convex polygon edges; The integer N is the number of grid points in each dimension; An ordered pair of real numbers (a_k, b_k) indicates the coefficients of a line l^k which is the extension of an edge of the convex polygon; The binary variable $l_{ij}^k = 1$ if and only if the grid point $\boldsymbol{g}_{ij}$, indexed by (i,j), lies on the specified side of the line l^k; The binary variable $z_{ij} = 1$ if and only if the grid point $\boldsymbol{g}_{ij}$ lies inside the polygon. We will fully discuss the roles of these decision variables in what follows.

Our goal is to make the convex polygon as small as possible without violating the constraints. This can be realized by minimizing the number of grid points lying inside the polygon. Formally speaking, our optimization objective function can be formulated as

$$\min \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij}, \tag{7}$$

where $\sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij}$ is a linear function of binary variables z_{ij}.

B. Constraints on Continuous Variables

All the planar lines not passing through the origin can be represented by

$$\{ax + by - 1 = 0 : a, b \in \mathbb{R} \wedge a^2 + b^2 \neq 0\},$$

and different pairs of coefficients (a,b) determine different lines.

A line $l^k := a_k x + b_k y - 1 = 0$ always divides the entire plane into two half-planes. We refer to the half plane $\{(x,y) : a_k x + b_k y - 1 < 0\}$ where the origin $(0,0)$ lies as the *strict inner side* of the line, and $\{(x,y) : a_k x + b_k y - 1 \leq 0\}$ as the *inner side* of the line. A strict inner side is a proper subset of the inner side of the same line.

For any $n \geq 3$ planar lines not passing through the origin, the region S bounded by these lines is defined as

$$S := \{(x,y) : \bigwedge_{k=1}^{n} a_k x + b_k y - 1 \leq 0\}, \tag{8}$$

where $a_k, b_k \in \mathbb{R}$, $a_k^2 + b_k^2 \neq 0$, and $n \geq 3$. On the one hand, since S is the intersection of n inner sides, then S is a convex set and the origin $(0, 0) \in S$; On the other hand, the region S may be not a bounded set (In other words, the boundary of S may be not enclosing). Even though bounded, the boundary of S may be a polygon with the number of edges less than n, or even not a polygon but an object consisting of line segments or a point only.

We look for conditions that ensure S is a bounded n sided convex polygon with the origin in its interior. To do this, we first introduce the following concepts.

Definition 4 (Digraph [34])**.** A finite directed graph (in short, a *digraph*) is a 2-tuple (V, E), where $V \subseteq \mathbb{R}^2$ is a finite set of planar points called nodes and $E \subseteq V \times V$ is a set of ordered pairs of nodes called (directed) edges. Self-loops are allowed in a digraph. For $u, v \in V$, if the ordered pair $(u, v) \in E$, it denotes an edge from u to v.

Definition 5 (Path [34])**.** A finite directed path (in short, a *path*) given by a finite sequence of planar points $S_g := (S_g[0], \ldots, S_g[m])$ where $S_g[i] \in \mathbb{R}^2, i \in \{0, \ldots, m\}$, is a digraph whose set of nodes $V = \{S_g[i] : i \in \{0, \ldots, m\}\}$ and set of edges $E = \{(S_g[i], S_g[i+1]) : i \in \{0, \ldots, m-1\}\}$. In a path S_g, one node $S_g[i]$ *immediately precedes* another node $S_g[j]$ if and only if there is an edge $(S_g[i], S_g[j])$; *Two nodes are consecutive* if and only if one immediately precedes the other; *Three nodes are consecutive* if and only if the first one immediately precedes the second and the second immediately precedes the third.

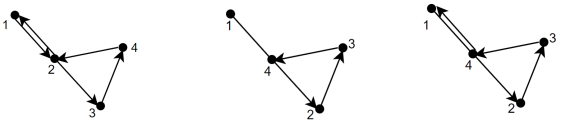
Note that if $S_g[i] = S_g[j], i \neq j$, then $\{S_g[i], S_g[j]\} = \{S_g[i]\}$, according to the Axiom of Extensionality in axiomatic set theory [35]. For a node $S_g[i]$ in S_g, the node which $S_g[i]$ immediately precedes (resp. which immediately precedes $S_g[i]$) may not exist, and even if it exists, it may be not unique. If there is an edge $(S_g[i], S_g[i])$, then $S_g[i]$ immediately precedes itself. It is possible that $S_g[i]$ immediately precedes $S_g[j]$ and $S_g[j]$ immediately precedes $S_g[i]$ at the same time if $(S_g[i], S_g[j])$ and $(S_g[j], S_g[i])$ are both edges in S_g.

Definition 6 (Graph and Undirected Version of Digraph [34])**.** A finite undirected graph (in short, a *graph*) is a 2-tuple (V, E), where $V \subseteq \mathbb{R}^2$ is a finite set of planar points called nodes and E is a set of unordered pairs of nodes called (undirected) edges. Self-loops are not allowed in a graph. For $u, v \in V, u \neq v$, if the set $\{u, v\} \in E$, it denotes an edge between u and v. A graph (V, E) is the *undirected version* of a digraph (V, E') if and only if $(u, v) \in E' \vee (v, u) \in E' \iff \{u, v\} \in E$ for $u, v \in V, u \neq v$.

In this paper, these definitions emphasize the planar geometric properties. For example, the positions of nodes are 2D coordinates; The shape of an edge connecting two distinct nodes is a straight line segment and the shape of a self-loop is a point. The operation of taking undirected version preserves geometric properties. For example, for any three nodes collinear in a digraph, these nodes must also be collinear in the graph which is the undirected version of the digraph.

Definition 7 (Enclosing Path)**.** A path S_g is *enclosing* if and only if: 1) There are $n \geq 3$ different nodes; 2) No nodes appear more than once in the sequence S_g except for the case of the initial and final node, which should be identical; 3) For any edge connecting two nodes, no other nodes lie on that edge.

For example, all the paths in Fig. 3 are not enclosing. The path $(1, 2, 3, 4, 2, 1)$ in Fig. 3a violates Condition 2) in Definition 7 because Node 2 appears more than once; The path $(1, 2, 3, 4)$ in Fig. 3b also violates Condition 2) because the first node 1 and last node 4 are not identical; The path $(1, 2, 3, 4, 1)$ in Fig. 3c satisfies first two conditions, but violates Condition 3) because Node 4 is on the edge $(1, 2)$.



(a) A path violating Condition 2) (b) A path violating Condition 2) (c) A path violating Condition 3)

Fig. 3. Three cases of paths that are not enclosing

Through the definition of an enclosing path, we can introduce a definition of an enclosing graph.

Definition 8 (Enclosing Graph)**.** A graph G is *enclosing* if and only if there is an enclosing path S_g whose undirected version is G.

For example, the graph G in Fig. 4 is not enclosing because any path whose undirected version is G is not an enclosing path.

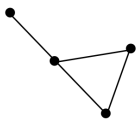


Fig. 4. A graph that is not enclosing

It is straightforward to see that if a path S_g is enclosing, then its undirected version G is also enclosing. Conversely, if a path is not enclosing, its undirected version may still be enclosing.

Inspired by Definition 7 but subtly different from it, we define the following notion.

Definition 9 (Formally Enclosing Sequence of Symbols)**.** Let S_s be a finite sequence of symbols. The sequence S_s is *formally enclosing* if and only if: 1) There are $n \geq 3$ formally different symbols; 2) No symbols appear more than once except for the case in which the initial and final symbol are formally identical.

In a finite sequence of symbols S_s, one symbol *immediately precedes* another symbol if and only if there is an appearance $S_s[i]$ of the first one and an appearance $S_s[j]$ of the second such that $j = i + 1$; *Two symbols are consecutive* if and only if one immediately precedes the other; *Three symbols are*

consecutive if and only if the first one immediately precedes the second and the second immediately precedes the third.

A finite sequence of symbols is just an ordered structure, which neither has to be enclosing nor has geometric meaning. To address this, we introduce an isomorphism map, as defined below.

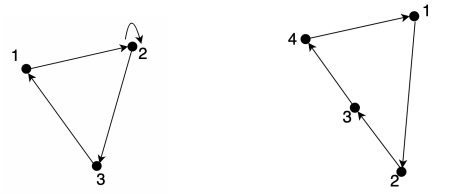
Definition 10 (Isomorphism Type A)**.** Let f_{tag} be a map from a finite sequence of symbols S_s to a path S_g. The map f_{tag} is an *isomorphism type A* if and only if the following properties hold for f_{tag}: 1) **Injective**: formally different symbols in S_s refer to different nodes in S_g; 2) **Surjective**: every node in S_g is the image of a symbol in S_s; 3) **Order-Preserving**: for any two symbols $S_s[i], S_s[j]$ in S_s, the symbol $S_s[i]$ immediately precedes $S_s[j]$ if and only if $(f_{\text{tag}}(S_s[i]), f_{\text{tag}}(S_s[j]))$ is an edge in S_g; 4) **Not on the edge**: for any two consecutive symbols $S_s[i], S_s[i+1]$ and any third symbol $S_s[k]$ in S_s, if $S_s[k]$ is formally different from $S_s[i]$ and $S_s[i+1]$, then $f_{\text{tag}}(S_s[k])$ is not on the edge $(f_{\text{tag}}(S_s[i]), f_{\text{tag}}(S_s[i+1]))$ in S_g.

Remark 1. Observe that if there is an isomorphism type A f_{tag} from a finite sequence of symbols S_s to a path S_g, then S_s is formally enclosing if and only if S_g is enclosing. We have seen that if S_g is enclosing, then its undirected version G is also enclosing. Therefore, we conclude that if f_{tag} is an isomorphism type A from S_s to S_g, and S_s is formally enclosing, then the graph G, as the undirected version of S_g, is also enclosing.

In addition to the discussions of enclosing paths and graphs above, we would also like to explore under which conditions it suffices to guarantee non-degenerate paths and graphs.

Definition 11 (Non-Degenerate Path and Graph)**.** Let S_g (resp. G) be a path (resp. a graph) with nodes given by a set of $n \geq 3$ points in $\mathbb{R}^2$. The path S_g is *a non-degenerate path* if and only if any three consecutive nodes in the sequence S_g are not collinear. The graph G is *a non-degenerate graph* if and only if there is a non-degenerate path S_g whose undirected version is G. A path S_g (resp. a graph G) is *degenerate* if and only if it is not non-degenerate.

Requiring that the number of nodes is $n = 4$, the following Fig. 5 gives two degenerate paths. In Fig. 5a, the path $(1, 2, 2, 3, 1)$ is degenerate, because the number of nodes in the path is three less than $n = 4$ and there are three consecutive nodes $(1, 2, 2)$ that are collinear. The path $(1, 2, 3, 4, 1)$ in Fig. 5b is also degenerate, because there are three consecutive nodes $(2, 3, 4)$ that are collinear.



(a) A degenerate path $(1, 2, 2, 3, 1)$ with three nodes

(b) A degenerate path $(1, 2, 3, 4, 1)$ with four nodes

Fig. 5. Two cases of degenerate paths

Requiring that the number of nodes is $n = 4$, the following Fig. 6 gives two degenerate graphs, which are the undirected version of the two paths shown in Fig. 5, respectively.

(a) A degenerate graph with three nodes

(b) A degenerate graph with four nodes

Fig. 6. Two cases of degenerate graphs

Note that non-degenerate graphs (resp. paths) and enclosing graphs (resp. paths) are two independent concepts. The following Fig. 7 gives examples to illustrate the difference between those two concepts.

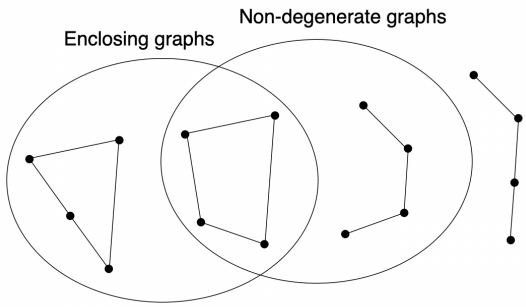


Fig. 7. Examples of enclosing or non-degenerate graphs

Definition 12 (Isomorphism Type B)**.** Let f_{tag} be a map from a finite sequence of symbols S_s to a path S_g. The map f_{tag} is an *isomorphism type B* if and only if the following properties hold for f_{tag}: 1) **Injective**: formally different symbols in S_s refer to different nodes in S_g; 2) **Surjective**: every node in S_g is the image of a symbol in S_s; 3) **Order-Preserving**: for any two symbols $S_s[i], S_s[j]$ in S_s, the symbol $S_s[i]$ immediately precedes $S_s[j]$ if and only if $(f_{\text{tag}}(S_s[i]), f_{\text{tag}}(S_s[j]))$ is an edge in S_g; 4) **Noncollinearity**: for any three consecutive symbols in S_s, their images are not collinear in S_g.

Remark 2. Observe that if there is an isomorphism type B f_{tag} from a finite sequence of symbols S_s to a path S_g, and there are $n \geq 3$ formally different symbols in S_s, then S_g and its undirected version G are both non-degenerate.

Note that Condition 4) in Definition 12 of isomorphism type B is different from Condition 4) in Definition 10 of isomorphism type A. For example, the label shown in Fig. 8a gives a map f_{tag} from the sequence of symbols (“a”, “b”, “c”, “d”, “b”) to the path (a, b, c, d, b). The map f_{tag} is an isomorphism type A but not an isomorphism type B, because there are three consecutive symbols “a”, “b”, “c” whose images a, b, c are collinear in the path. This violates Condition 4) in Definition 12. In contrast, the label shown in

Fig. 8b gives another map f'_{tag} from the sequence of symbols ("a", "b", "c", "d") to the path (a, b, c, d). The map f'_{tag} is an isomorphism type B but not an isomorphism type A, since although the symbol "d" is formally different from each of the two consecutive symbols "a", "b", Node d is on the edge (a, b) in the path. This violates Condition 4) in Definition 10.

(a) An isomorphism type A

(b) An isomorphism type B

Fig. 8. Examples of isomorphisms type A or type B

We go back to look for the conditions that ensure S in Eq. (8) is a bounded n sided convex polygon with the origin in its interior.

For any two lines $l^i := a_i x + b_i y - 1 = 0$ and $l^j := a_j x + b_j y - 1 = 0$, if $D_{ij} := a_i b_j - b_i a_j \neq 0$, then there is a unique intersection point between those two lines, which is

$$V_{ij} := l^i \cap l^j = \left(-\frac{b_i - b_j}{D_{ij}}, \frac{a_i - a_j}{D_{ij}} \right).$$

Note that $V_{ij} = V_{ji}$. The fact $D_{ij} \neq 0$ implies that $a_i^2 + b_i^2 \neq 0 \wedge a_j^2 + b_j^2 \neq 0$, and also implies that $a_i \neq a_j \vee b_i \neq b_j$, namely, $V_{ij} \neq (0, 0)$. This makes sense since neither lines passes through the origin and therefore the origin is definitely not the intersection point. The sign of D_{ij} depends on the order of i and j, namely, $D_{ij} = -D_{ji}$. For two lines l^i, l^j sharing a unique intersection point, if we can rotate l^i's normal vector (a_i, b_i) to l^j's normal vector (a_j, b_j) counterclockwise about the origin by an angle $\theta \in (0, \pi)$, then $D_{ij} > 0$; otherwise, $D_{ij} < 0$. When $D_{ij} < 0$, we can swap the value of (a_i, b_i) with that of (a_j, b_j) so that $D_{ij} > 0$. Hence, without loss of generality, we can assume that $D_{ij} > 0$.

For simplicity of notation, we define a unary operation $i^{\oplus}$ for $i \in \{1, \ldots, n\}$

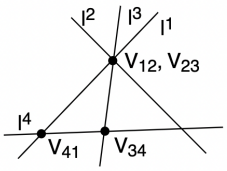
$$i^{\oplus} := \begin{cases} i + 1, & i \in \{1, \ldots, (n-1)\}, \\ 1, & i = n, \end{cases}$$

where $i + 1$ is the ordinary arithmetic operation of addition. Inversely, we can define another unary operation $i^{\ominus}$ for $i \in \{1, \ldots, n\}$, that is, $j = i^{\ominus}$ if and only if $j^{\oplus} = i$.

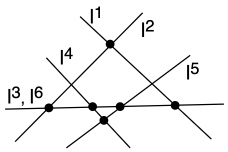
Consider $n \geq 3$ planar lines not passing through the origin, denoted by $l^k,\ k \in \{1, \ldots, n\}$. We hope the boundary of the region S, generated by these lines in the way Eq. (8) gives, is an enclosing n sided convex polygon with the origin in its interior. A necessary condition to make it is that there are at least n different line intersection points serving as n nodes of the polygon. Without loss of generality, we require the existence of such n formally different intersection points $V_{ij},\ i \in \{1, \ldots, n\}, j = i^{\oplus}$ (which also implicitly requires $l^i \neq l^j,\ i \in \{1, \ldots, n\}, j = i^{\oplus}$). To enforce this, we formulate the following constraints

$$D_{ij} := a_i b_j - b_i a_j > 0, \quad i \in \{1, \ldots, n\},\ j = i^{\oplus}. \tag{9}$$

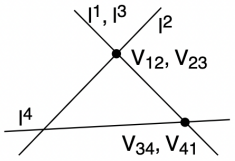
Nevertheless, satisfying Eq. (9) can not ensure that there are indeed n different intersection points (resp. lines), because two formally different representations of the intersection points (resp. lines) may refer to an identical intersection point (resp. line), as illustrated in Fig. 9.



(a) The point V_{12} and V_{23} are identical



(b) The line l^3 and l^6 are identical

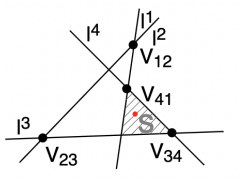


(c) The line l^1 and l^3, the point V_{12} and V_{23}, as well as the point V_{34} and V_{41} are identical

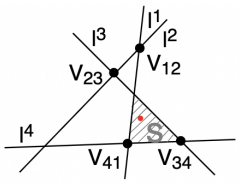
Fig. 9. Scenarios of identical intersection points or lines while satisfying Eq. (9)

The boundary of a region S generated by a finite set of lines can be defined as the undirected version G of a path S_g. Throughout the rest of this paper, when a graph G is the undirected version of a path S_g, we say the graph G is *formed* by S_g. For any $n \geq 3$ planar lines $l^k,\ k \in \{1, \ldots, n\}$ not passing through the origin, if there are n formally different intersection points $V_{ij},\ i \in \{1, \ldots, n\}, j = i^{\oplus}$, we can construct a path given by a finite sequence of intersection points $S_g := (V_{12}, V_{23}, \ldots, V_{(n-1)n}, V_{n1}, V_{12})$. If the graph G formed by S_g is an enclosing n sided convex polygon with the origin in its interior, then the boundary of the region S is identical to the graph G, and therefore the boundary of the region S is also an enclosing n sided convex polygon with the origin in its interior. Hence, to make the boundary of the region S an enclosing n sided convex polygon with the origin in its interior, we just need to ensure the graph G formed by S_g is an enclosing n sided convex polygon with the origin in its interior.

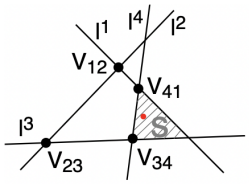
However, for certain $n \geq 3$ planar lines not passing through the origin that have n formally different intersection points $V_{ij},\ i \in \{1, \ldots, n\}, j = i^{\oplus}$, the graph G formed by S_g may be not an enclosing n sided convex polygon with the origin in its interior. Specifically, 1) the graph G may be not enclosing. For example, the graph G in Fig. 9c is a line segment and is not enclosing; 2) The graph G may degenerate to a polygon whose number of edges less than n, or even not a polygon but line segments or a point. For example, the graph G in Fig. 9a is a triangle, not a quadrilateral as required; 3) The graph G may be not convex (may be concave or self-intersecting), as illustrated in Fig. 10a and Fig. 10b; 4) The origin $(0, 0)$ may be not in the interior of the graph G, as shown in Fig. 10c.



(a) A concave quadrilateral formed by $(V_{12}, V_{23}, V_{34}, V_{41}, V_{12})$



(b) A self-intersecting quadrilateral formed by $(V_{12}, V_{23}, V_{34}, V_{41}, V_{12})$



(c) A convex quadrilateral formed by $(V_{12}, V_{23}, V_{34}, V_{41}, V_{12})$ with the origin in its exterior

Fig. 10. Cases of graphs that are not enclosing n sided convex polygons with the origin in its interior (The shadowed area is the region S defined in Eq. (8) and the red point is the origin)

As discussed above, the constraints Eq. (9) are not sufficient to guarantee that the graph G formed by S_g is an enclosing n sided convex polygon with the origin in its interior. To this end, we provide additional constraints in Theorem 1.

Before proceeding to Theorem 1, we first introduce Lemma 1. This lemma is useful in the sense that once constraints Eq. (9) and Eq. (10) are satisfied, we no longer need to explicitly enforce that all n intersection points (resp. lines) are indeed different from each other.

Lemma 1. *Assume that there are $n \geq 3$ planar lines not passing through the origin $(0,0)$, denoted by $l^k := a_k x + b_k y - 1 = 0, k \in \{1, \ldots, n\}$. If there are n intersection points $V_{ij} := l^i \cap l^j,\ i \in \{1, \ldots, n\}, j = i^{\oplus}$, and the coefficients of these lines satisfy the following inequalities*

$$\begin{aligned} &-a_k\,(b_i - b_j) + b_k\,(a_i - a_j) - (a_i b_j - b_i a_j) < 0, \\ &i \in \{1, \ldots, n\}, j = i^{\oplus}, k \in \{1, \ldots, n\} - \{i, j\}, \end{aligned} \tag{10}$$

then, any two formally different representations of intersection points (resp. lines) refer to different intersection points (resp. lines), and therefore all n intersection points (resp. lines) are different from each other.

Proof. See Section VIII-A. ■

Next, the following Theorem 1 gives the conditions which suffice to ensure the boundary of the region S in Eq. (8) is an enclosing n sided convex polygon with the origin in its interior. Intuitively, constraints Eq. (10) check each intersection point $V_{ij}, i \in \{1, \ldots, n\}, j = i^{\oplus}$ against all its opposite $(n-2)$ lines $l^k, k \in \{1, \ldots, n\} - \{i, j\}$.

Theorem 1. *Assume that there are $n \geq 3$ planar lines not passing through the origin $(0,0)$, denoted by $l^k := a_k x + b_k y - 1 = 0,\ k \in \{1, \ldots, n\}$. If there are n intersection points $V_{ij} := l^i \cap l^j,\ i \in \{1, \ldots, n\}, j = i^{\oplus}$, and the coefficients of these lines satisfy Eq. (10), then the boundary of the region $S := \{(x,y) : \bigwedge_{k=1}^{n} a_k x + b_k y - 1 \leq 0\}$ generated by these lines is the graph G formed by the sequence of intersection points $S_g := (V_{12}, V_{23}, \ldots, V_{(n-1)n}, V_{n1}, V_{12})$, which defines an enclosing n sided convex polygon with the origin in its interior.*

Proof. See Section VIII-B. ■

So far, we have just derived constraints Eq. (9) and Eq. (10) to ensure that the polygon of interest contains the origin $(0,0)$. The following result extends the applicability of these conditions to contain an arbitrary point.

Corollary 1. *Assume that there are $n \geq 3$ planar lines not passing through the point $(\bar{x}, \bar{y})$, denoted by $l^k := a_k(x - \bar{x}) + b_k(y - \bar{y}) - 1 = 0,\ k \in \{1, \ldots, n\}$. If constraints Eq. (9) and Eq. (10) hold, then the boundary of the region $S := \{(x,y) : \bigwedge_{k=1}^{n} a_k(x-\bar{x}) + b_k(y-\bar{y}) - 1 \leq 0\}$ generated by these lines is the graph G formed by the sequence of intersection points $S_g := (V_{12}, V_{23}, \ldots, V_{(n-1)n}, V_{n1}, V_{12})$, which defines an enclosing n sided convex polygon with the point $(\bar{x}, \bar{y})$ inside.*

Proof. See Section VIII-C. ■

We finally conclude that, according to Corollary 1, Eq. (9) and Eq. (10) are the constraints that we formulate for $2n$ continuous variables $a_k, b_k \in \mathbb{R},\ k \in \{1, \ldots, n\}$. For any $n \geq 3$ planar lines not passing through an arbitrary point, if satisfying Eq. (9) and Eq. (10), we can guarantee that the boundary of the region $S := \{(x,y) : \bigwedge_{k=1}^{n} a_k(x - \bar{x}) + b_k(y - \bar{y}) - 1 \leq 0\}$ generated by these lines is an enclosing n sided convex polygon with that point in its interior.

C. Constraints Relating Continuous and Discrete Variables

Recall that, a mesh of grid points is a map $\boldsymbol{g}$ given by $(x_i, y_j) = \boldsymbol{g}_{ij},\ i, j \in \{1, \ldots, N\}$ as defined in Eq. (3), and we have obtained a normalized weight matrix $\boldsymbol{w} := [w_{ij}]$ such that every grid point $\boldsymbol{g}_{ij},\ i, j \in \{1, \ldots, N\}$ has a normalized weight w_{ij}, respectively, in Eq. (6). The index of the grid point whose normalized weight is the greatest is denoted by $(\bar{i}, \bar{j}) := \arg\max_{i,j}(\boldsymbol{w})$, and its coordinate is $(x_{\bar{i}}, y_{\bar{j}}) := \boldsymbol{g}_{\bar{i}\bar{j}}$, which plays the role of $(\bar{x}, \bar{y})$ in Section IV-B.

The following binary variables $z_{ij} \in \{0, 1\}$ are defined so that $z_{ij} = 1$ if and only if the grid point $\boldsymbol{g}_{ij}$ lies inside the polygon obtained in Section IV-B. In other words,

$$z_{ij} = 1 \implies \bigwedge_{k=1}^{n} a_k(x_i - x_{\bar{i}}) + b_k(y_j - y_{\bar{j}}) - 1 \leq 0, \tag{11}$$
$$i, j \in \{1, \ldots, N\},$$

$$z_{ij} = 0 \implies \bigvee_{k=1}^{n} a_k(x_i - x_{\bar{i}}) + b_k(y_j - y_{\bar{j}}) - 1 > 0, \tag{12}$$
$$i, j \in \{1, \ldots, N\}.$$

To convert the logical constraints above into equivalent algebraic constraints, we introduce new binary variables

$l_{ij}^k \in \{0,1\}$ such that $l_{ij}^k = 1$ if and only if the grid point $\boldsymbol{g}_{ij}$ lies on the inner side of the line l^k. That is,

$$l_{ij}^k = 1 \implies a_k(x_i - x_{\bar{i}}) + b_k(y_j - y_{\bar{j}}) - 1 \leq 0, \quad i,j \in \{1,\dots,N\}, \quad k \in \{1,\dots,n\},$$

$$l_{ij}^k = 0 \implies a_k(x_i - x_{\bar{i}}) + b_k(y_j - y_{\bar{j}}) - 1 > 0, \quad i,j \in \{1,\dots,N\}, \quad k \in \{1,\dots,n\},$$

where $n \in \{x \in \mathbb{N} : x \geq 3\}$. These can be formulated as algebraic constraints via big-M representation [36] as

$$a_k(x_i - x_{\bar{i}}) + b_k(y_j - y_{\bar{j}}) - 1 \leq M_{ijk}^1(1 - l_{ij}^k) \quad i,j \in \{1,\dots,N\}, \quad k \in \{1,\dots,n\}, \tag{13}$$

$$-a_k(x_i - x_{\bar{i}}) - b_k(y_j - y_{\bar{j}}) + 1 < M_{ijk}^2 l_{ij}^k \quad i,j \in \{1,\dots,N\}, \quad k \in \{1,\dots,n\}, \tag{14}$$

where the parameters $M_{ijk}^1, M_{ijk}^2,\ i,j \in \{1,\dots,N\},\ k \in \{1,\dots,n\}$ are sufficiently large constants. They are often set specifically across different constraints, but in this paper, we just need to choose a common constant for them.

The logical relationship between z_{ij} and l_{ij}^k is

$$z_{ij} = 1 \implies \sum_{k=1}^{n} l_{ij}^k = n \quad i,j \in \{1,\dots,N\},$$

$$z_{ij} = 0 \implies \sum_{k=1}^{n} l_{ij}^k \leq (n-1) \quad i,j \in \{1,\dots,N\},$$

and these can be formulated as algebraic constraints

$$\sum_{k=1}^{n} l_{ij}^k \geq n z_{ij} \quad i,j \in \{1,\dots,N\}, \tag{15}$$

$$\sum_{k=1}^{n} l_{ij}^k \leq (n-1) + z_{ij} \quad i,j \in \{1,\dots,N\}, \tag{16}$$

where $n \in \{x \in \mathbb{N} : x \geq 3\}$.

In this way, we have converted the original logical constraints Eq. (11) and Eq. (12), into equivalent algebraic inequalities Eq. (13), Eq. (14), Eq. (15) and Eq. (16). These constraints give the relation between $2n$ continuous variables $a_k, b_k \in \mathbb{R},\ k \in \{1,\dots,n\}$ and $N^2 n$ binary variables $l_{ij}^k \in \{0,1\},\ i,j \in \{1,\dots,N\},\ k \in \{1,\dots,n\}$, N^2 binary variables $z_{ij} \in \{0,1\},\ i,j \in \{1,\dots,N\}$.

D. Constraints on Discrete Variables

We would like to enforce that the grid point $\boldsymbol{g}_{\bar{i}\bar{j}} := (x_{\bar{i}}, y_{\bar{j}})$ lies inside the polygon obtained in Section IV-B. Recall that, the binary variable $z_{ij} = 1$ if and only if the grid point $\boldsymbol{g}_{ij}$ lies inside the polygon. Therefore, we impose that

$$z_{\bar{i}\bar{j}} = 1. \tag{17}$$

In fact, Eq. (17) is unnecessary as $(x_{\bar{i}}, y_{\bar{j}}) \in S := \{(x,y) : \bigwedge_{k=1}^{n} a_k(x - x_{\bar{i}}) + b_k(y - y_{\bar{j}}) - 1 \leq 0\}$ given by Corollary 1 in Section IV-B, which means this grid point must lie inside the polygon. However, we explicitly list it here for better clarity.

The convex polygon is intended to approximate the PRS. To guarantee a level of confidence of the PRS, we impose that the sum of the normalized weights w_{ij} of the grid points which lie inside the polygon should be greater than this confidence level α. That is,

$$\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} z_{ij} \geq \alpha. \tag{18}$$

Eq. (17) and Eq. (18) lead to two constraints that we formulate for N^2 binary variables $z_{ij} \in \{0,1\},\ i,j \in \{1,\dots,N\}$.

E. Formulation of MINLP Optimization Framework

To solve the problem described in Section IV-A, we have chosen the objective function $\min \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij}$ in Eq. (7), and introduced the following three types of decision variables with $n \in \{x \in \mathbb{N} : x \geq 3\}$ in Section IV-A:

- $2n$ continuous variables $a_k, b_k \in \mathbb{R},\ k \in \{1,\dots,n\}$;
- $N^2 n$ binary variables $l_{ij}^k \in \{0,1\},\ i,j \in \{1,\dots,N\},\ k \in \{1,\dots,n\}$;
- N^2 binary variables $z_{ij} \in \{0,1\},\ i,j \in \{1,\dots,N\}$.

Collect the previous constraints Eq. (9), Eq. (10) from Section IV-B; Eq. (13), Eq. (14), Eq. (15), Eq. (16) from Section IV-C; Eq. (17), Eq. (18) from Section IV-D.

We formally formulate the problem stated inSection II as an MINLP optimization framework below. All coordinates are expressed in a global coordinate system. If not specified, $\forall i,j \in \{1,\dots,N\},\ \forall k \in \{1,\dots,n\}$.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij} \\
\text{s.t.} \quad & a_i b_j - b_i a_j > 0, \quad \forall i \in \{1,\dots,n\}, j = i^{\oplus}; \\
& -a_k\,(b_i - b_j) + b_k\,(a_i - a_j) < a_i b_j - b_i a_j, \quad \forall i \in \\
& \{1,\dots,n\},\ j = i^{\oplus},\ \forall k \in \{1,\dots,n\} - \{i,j\}; \\
& a_k(x_i - x_{\bar{i}}) + b_k(y_j - y_{\bar{j}}) - 1 \le M^1_{ijk}(1 - l^k_{ij}), \quad \forall i,j,k; \\
& -a_k(x_i - x_{\bar{i}}) - b_k(y_j - y_{\bar{j}}) + 1 < M^2_{ijk} l^k_{ij}, \quad \forall i,j,k; \\
& \sum_{k=1}^{n} l^k_{ij} \ge n z_{ij}, \quad \forall i,j; \\
& \sum_{k=1}^{n} l^k_{ij} \le (n-1) + z_{ij}, \quad \forall i,j; \\
& z_{\bar{i}\bar{j}} = 1; \\
& \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} z_{ij} \ge \alpha; \\
& a_k, b_k \in \mathbb{R}, \quad \forall k; \\
& l^k_{ij} \in \{0,1\}, \quad \forall i,j,k; \\
& z_{ij} \in \{0,1\}, \quad \forall i,j.
\end{aligned}
\tag{19}
$$

V. Solution Method

In this section, we develop a heuristic algorithm to efficiently solve the previous MINLP optimization.

A. MINLP Optimal Algorithm

Although the optimization framework Eq. (19) is an MINLP problem with nonlinear constraints, all constraints except for Eq. (9) and Eq. (10) are in fact linear ones. Fortunately, even the quadratic constraints Eq. (9) and Eq. (10) are bilinear constraints that only involve the product of disjoint pairs of variables. Gurobi is well-suited for addressing such constraints. It employs cutting planes and branching algorithm; see [37] for more information. Leveraging this solver, **Algorithm 2** finds an optimal solution to Eq. (19), which results in a convex approximation of the PRS computed by Algorithm 1.

B. MINLP Heuristic Algorithm

However, Algorithm 2 is computationally expensive and scales poorly with a large number of grid points. To address this issue, we develop **Algorithm 3** that can efficiently solve Eq. (19) while ensuring accuracy. Algorithm 2 just serves as a benchmark that provides the optimal solution to Eq. (19) which is used only for comparison purposes. As explained in Fig. 11, this algorithm performs weighted sampling based on the KDE values to select representative grid points $\boldsymbol{g}'$ from all grid points $\boldsymbol{g}$. Unlike the original MINLP problem, which uses all grid points $\boldsymbol{g}$, a new MINLP problem can be formulated using representative grid points $\boldsymbol{g}'$. Then, the cutting planes and branching algorithm built in Gurobi can be applied to solve this newly formulated MINLP problem. As illustrated in Fig. 12, the optimal solution to the new MINLP problem is an approximation to the solution of the original MINLP problem. By doing so, efficiency comes at the cost of optimality. Since the size of $\boldsymbol{g}'$ can be far less than $\boldsymbol{g}$, the new MINLP problem reduces the number of decision variables and constraints, greatly contributing to reducing computational time.

Algorithm 2 MINLP Optimal Algorithm

1: **function** MINLPSolver(α, $\boldsymbol{g}$, $\boldsymbol{w}$)
2: $\bar{i}$, $\bar{j}$ = $\arg\max(\boldsymbol{w})$
3: Determine $x_{\bar{i}}$, $y_{\bar{j}}$ according to $\bar{i}$, $\bar{j}$ and $\boldsymbol{g}$
4: Formulate Eq. (19) for grid points $\boldsymbol{g}$ with weights $\boldsymbol{w}$ given confidence level α
5: Implement cutting planes and branching algorithm to solve Eq. (19) for a_k, b_k
6: **return** a_k, b_k, $x_{\bar{i}}$, $y_{\bar{j}}$
7: **function** FindPolygon($\boldsymbol{g}$, $\boldsymbol{w}$, α)
8: a_k, b_k, $x_{\bar{i}}$, $y_{\bar{j}}$ = MINLPSolver(α, $\boldsymbol{g}$, $\boldsymbol{w}$)
9: **return** a_k, b_k, $x_{\bar{i}}$, $y_{\bar{j}}$
10: FindPolygon($\boldsymbol{g}$, $\boldsymbol{w}$, α)

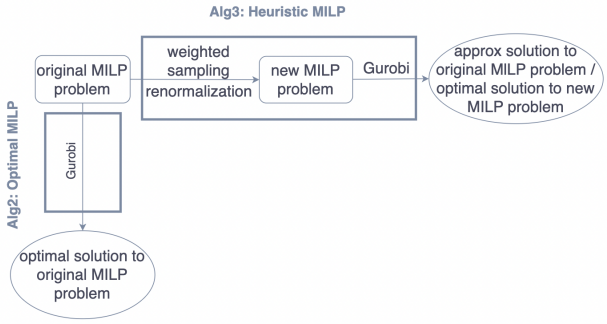


Fig. 11. Procedural difference between MINLP Heuristic and MINLP Optimal

In the pseudo-code of Algorithm 3, we use N_s to represent the number of representative grid points $\boldsymbol{g}'$. From Lines 1 through 6, we use the AES algorithm proposed by [38] to randomly select N_s representative grid points $\boldsymbol{g}'$ from all N^2 grid points $\boldsymbol{g}$ without replacement, according to their normalized weights $\boldsymbol{w}$. Recall that $\boldsymbol{g}$ is the output of Algorithm 1 and $\boldsymbol{w} := [w_{ij}]$ is obtained through Eq. (6). From Line 7 through 9, the original weights $\boldsymbol{w}'$ of representative grid points $\boldsymbol{g}'$ are re-normalized to $\hat{\boldsymbol{w}}'$, so that their sum is equal to one. The process of re-normalization is crucial to guarantee the accuracy of the solution obtained from Algorithm 3, when

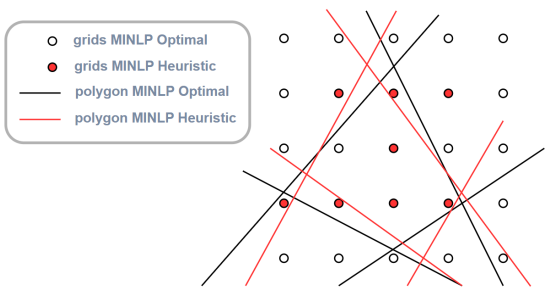


Fig. 12. Results difference between MINLP Heuristic and MINLP Optimal

compared with the solution after implementing Algorithm 2.

Algorithm 3 MINLP Heuristic Algorithm

function WGTSAMP($\boldsymbol{g}$, $\boldsymbol{w}$, N_s)
  $u_{ij} = \text{random}(0, 1)$
  $k_{ij} = u_{ij}^{1/w_{ij}}$
  Take first N_s grids from $\boldsymbol{g}$ in descending order of k_{ij} as representative grids $\boldsymbol{g}'$
  Take the weights of those N_s grids from $\boldsymbol{w}$ as representative weights $\boldsymbol{w}'$
  return $\boldsymbol{g}'$, $\boldsymbol{w}'$

function NORMWGT($\boldsymbol{w}'$, N_s)
  $\hat{\boldsymbol{w}}' = \boldsymbol{w}' + \text{ones}(N_s) \cdot (1 - \text{sum}(\boldsymbol{w}'))/N_s$
  return $\hat{\boldsymbol{w}}'$

function MINLPSOLVER($\boldsymbol{g}'$, $\hat{\boldsymbol{w}}'$, α, $\boldsymbol{g}$)
  $\bar{i}$, $\bar{j} = \arg\max(\hat{\boldsymbol{w}}')$
  Determine $x_{\bar{i}}$, $y_{\bar{j}}$ according to $\bar{i}$, $\bar{j}$ and $\boldsymbol{g}$
  Formulate Eq. (19) for grid points $\boldsymbol{g}'$ with weights $\hat{\boldsymbol{w}}'$ given confidence level α
  Implement cutting planes and branching algorithm to solve Eq. (19) for a_k, b_k
  return a_k, b_k, $x_{\bar{i}}$, $y_{\bar{j}}$

function FINDPOLYGON($\boldsymbol{g}$, $\boldsymbol{w}$, N_s, α)
  $\boldsymbol{g}'$, $\boldsymbol{w}'$ = WGTSAMP($\boldsymbol{g}$, $\boldsymbol{w}$, N_s)
  $\hat{\boldsymbol{w}}'$ = NORMWGT($\boldsymbol{w}'$, N_s)
  a_k, b_k, $x_{\bar{i}}$, $y_{\bar{j}}$ = MINLPSOLVER($\boldsymbol{g}'$, $\hat{\boldsymbol{w}}'$, α, $\boldsymbol{g}$)
  Determine the polygon S according to a_k, b_k, $x_{\bar{i}}$, $y_{\bar{j}}$
  return S

FINDPOLYGON($\boldsymbol{g}$, $\boldsymbol{w}$, N_s, α)

VI. Main Results

In this section, we conduct comprehensive case studies to compare the performance of the MINLP Optimal algorithm (Algorithm 2), and the MINLP Heuristic algorithm (Algorithm 3), with the Bounding Box algorithm introduced in [28], as well as the impact of different parameters on the performance of the algorithms. We also discuss the robustness of the MINLP Heuristic algorithm. The tests were implemented in Python 3.9 and on an Intel(R) Core(TM) i9-12900KF, 3187 Mhz, 16 Core(s), 24 Logical Processor(s) Desktop with 64GB RAM.

Case studies are divided into two stages:

1) Solution stage: According to KDE values and the PRS obtained from a collection of M data samples running Algorithm 1, we formulate an MINLP problem. Then, MINLP Optimal and MINLP Heuristics are implemented respectively to find convex approximations of the PRS. As a comparison, the Bounding Box algorithm is applied to find a convex quadrilateral bounding the PRS;

2) Testing stage: Given the obtained convex polygon in the previous stage, we can generate a new collection of N_{test} ($N_{\text{test}} \gg M$) data samples, and evaluate the ratio of the number of data samples that fall inside the polygon to the total number of data samples generated. As N_{test} increases, the ratio will converge to the true probability of the system state lying inside the polygon.

A. Cases Settings

1) Case I: In this case, we consider a Dubins vehicle model moving on a plane. The speed of the vehicle obeys a truncated Gaussian distribution $v \sim \mathcal{N}(\mu_1, \sigma_1)$ with $\mu_1 = 190\,\text{km/h}$ and $\sigma_1 = 5\,\text{km/h}$, and the speed falls inside the interval $[165, 220]\,\text{km/h}$. The heading angle of the vehicle obeys another truncated Gaussian distribution $\theta \sim \mathcal{N}(\mu_2, \sigma_2)$ with $\mu_2 = 10\,\text{degs}$ and $\sigma_2 = 30\,\text{degs}$, and the heading angle falls inside the interval $[-50, 70]\,\text{degs}$.

The position $\boldsymbol{x}_k := [x_k, y_k]^\top$ of the vehicle at time k can be modeled by a discrete-time dynamic system

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} + \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} v\Delta t,$$

where $\boldsymbol{x}_{k-1} := [x_{k-1}, y_{k-1}]^\top$ is the position of the vehicle at time $k-1$, and the speed v and heading angle θ are assumed to be constant during a time interval $\Delta t = 1\,\text{s}$. Due to the uncertainties arising from v and θ of the vehicle, the position $\boldsymbol{x}_k$ is a random vector that obeys a non-Gaussian distribution, which we call a fan-shaped distribution. The shadowed region in Fig. 13 indicates the reachable set of the position of the vehicle at a time point.

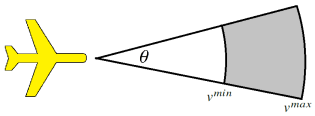


Fig. 13. Reachable set of the position of a vehicle at a time point

2) Case II: In this case, we display a scatter plot of the possible positions (data samples) $[x, y]^\top$ of a vehicle on the plane at a time point, which is generated by the marginal histograms for x and y respectively. As shown in Fig. 14, the position of the vehicle at that time point obeys a bimodal distribution which is non-Gaussian.

In the following, the bandwidth matrix $\mathbf{H}$ in Eq. (2) is set to $\left[\begin{smallmatrix} 0.2 & 0 \\ 0 & 0.2 \end{smallmatrix}\right]$, a common constant 10^4 is chosen for the parameters

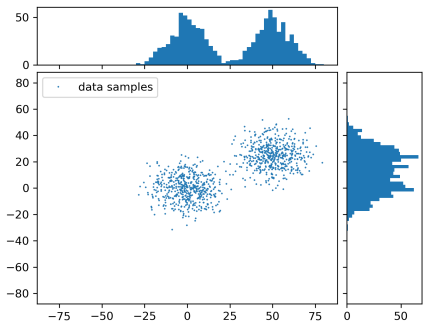


Fig. 14. Joint bimodal distribution generated by two marginal histograms

M^1_{ijk}, M^2_{ijk} in Eq. (13) and Eq. (14), and the number of data samples is $M = 1000$. The setting below is chosen as a baseline: The number of sides is $n = 4$, the confidence level is $\alpha = 90\%$, the number of grid points used by MINLP Optimal is $N^2 = 20^2$, and the number of grid points used by MINLP Heuristic is $N_s = 70$ for Case I and $N_s = 60$ for Case II. In the following figures and tables, the baseline is marked with the symbol "*".

B. Different Numbers of Sides

In this part, we compare the performance of different algorithms for different numbers of polygon sides: $n = 3, 4, 5$. The other parameters are the same as in the baseline.

The results are shown in Fig. 15 and Table I. Note that changing of the number of sides n is only applicable for MINLP Optimal and MINLP Heuristic. For MINLP Optimal or MINLP Heuristic, as n increases from 3 to 5, the area of the polygon becomes smaller, while the ratio is just slightly different and always closer to the confidence level α. This suggests that the results of both algorithms become less conservative while ensuring accuracy. However, the associated computational time increases as the number of decision variables and constraints in Eq. (19) increases with an increase on n. To achieve a balance between optimality and computational efficiency, we set $n = 4$.

When $n = 4$, the area and ratio of either MINLP Optimal or MINLP Heuristic are much smaller than those of the Bounding Box algorithm, which suggests that the polygons obtained by both MINLP Optimal and MINLP Heuristic algorithms are more tight and accurate than the Bounding Box. This clearly benefits real motion planning. When considering collision avoidance between the vehicle and an obstacle, a less conservative convex approximation of the PRS allows a larger feasible search area which may provide a better-planned trajectory. Plus, the computational time of MINLP Heuristic is far less than MINLP Optimal ($0.801\,\mathrm{s} < 70.5\,\mathrm{s}$ for Case I and $0.361\,\mathrm{s} < 18.6\,\mathrm{s}$ for Case II), while their results of area and ratio are still similar. This demonstrates the computational efficiency of MINLP Heuristic while ensuring accuracy, making online implementation feasible.

As stated in Section V, the polygon obtained by MINLP Optimal is optimal in the sense that it has a minimum area without violating constraints. However, for some cases in Table I, the area of MINLP Heuristic (e.g. $1807.9\,\mathrm{m}^2$) may be slightly less than MINLP Optimal (e.g. $1827.0\,\mathrm{m}^2$). The approximation error comes from the re-normalization with the newly selected N_s grid points $\boldsymbol{g}'$. The polygon obtained by MINLP Heuristic is an optimal solution to a new MINLP problem based on those selected N_s grid points, which is also an approximation to the original MINLP problem involving N^2 grid points. Since the original and new MINLP problems have different grid settings, there is a chance that the polygon area of MINLP Heuristic is less than MINLP Optimal. However, those errors are relatively small according to our experiments.

C. Different Numbers of Grid Points

Here, we compare the performance of the algorithms when considering different numbers of grid points: $N^2 = 10^2, 20^2, 30^2, 40^2$. For MINLP Heuristic, $N_s = 20, 70, 160, 280$ grid points are randomly selected for Case I and $N_s = 15, 60, 135, 240$ grid points for Case II, respectively. The other parameters are the same as in the baseline.

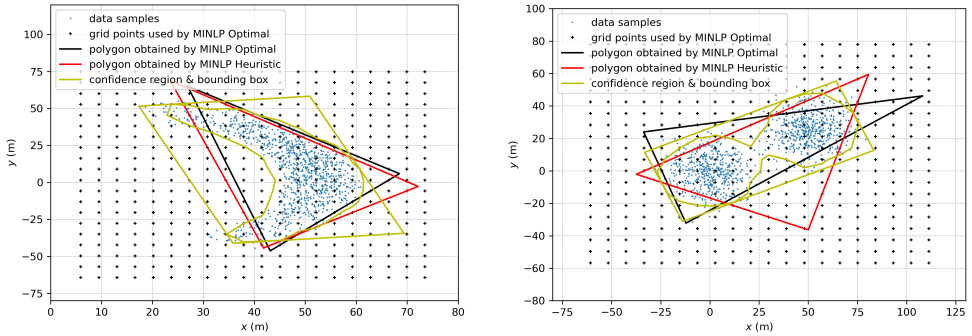
From Fig. 16 and Table II, we can draw some conclusions as follows. The results obtained by the implementation of all three algorithms given in Fig. 16a and Fig. 16b are visually very different from the other subfigures in Fig. 16. This is because when the number of grid points is too small ($N^2 = 10^2$), KDE can not approximate PDF well. As indicated in Table II, for every algorithm, as N^2 ranges from 20^2 to 40^2, the performance in ratio and area slightly improves. However, as N^2 increases, the increase in the number of decision variables and constraints leads to a significant increase in computational time. As shown in Table II, the computational time of MINLP Heuristic goes from $0.801\,\mathrm{s}$ to $125.5\,\mathrm{s}$ for Case I and from $0.361\,\mathrm{s}$ to $23.8\,\mathrm{s}$ for Case II. Thus, there is a slight improvement in accuracy and optimality, but at huge cost of efficiency. To reach a compromise, the number of grid points can be chosen as $N^2 = 20^2$.

When N^2 is fixed to 20^2, the area and ratio of MINLP Optimal and MINLP Heuristic are much smaller than those of the Bounding Box algorithm, indicating MINLP Optimal and MINLP Heuristic significantly outperform Bounding Box in terms of optimality and accuracy. Further, for MINLP Optimal and MINLP Heuristic, their results of ratio and area are quite similar. However, the computational time of MINLP Heuristic greatly outperforms MINLP Optimal, i.e., $0.801\,\mathrm{s} < 70.5\,\mathrm{s}$ for Case I and $0.361\,\mathrm{s} < 18.6\,\mathrm{s}$ for Case II, demonstrating the efficiency of MINLP Heuristic. Therefore, MINLP Heuristic can guarantee near-optimality, accuracy, and efficiency simultaneously, which makes it very well suited for real-time implementation.

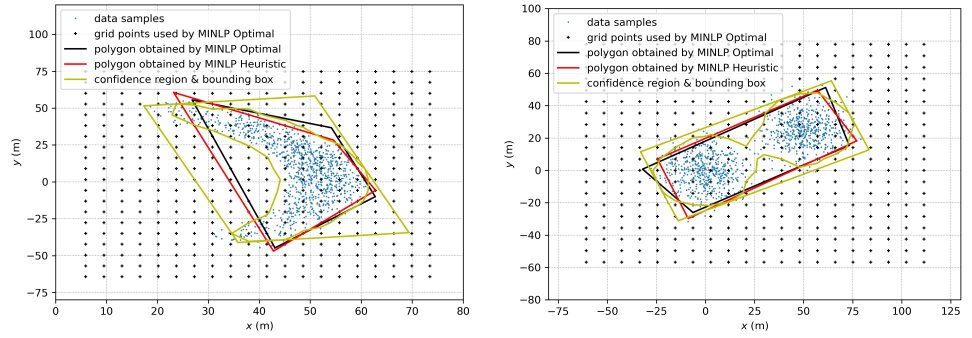
D. Different Confidence Levels

In this part, we compare the performance of the algorithms relative to different confidence levels: $\alpha = 90\%, 95\%$, and 99%. The other parameters are the same as in the baseline.

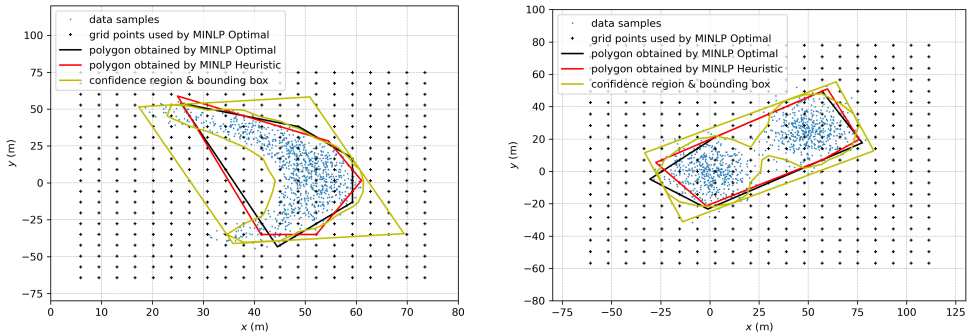
In Fig. 17 and Table III, for both Case I and II, as α increases, the area of the polygon obtained by MINLP Optimal, MINLP Heuristic or Bounding Box respectively increases while the computational time of each algorithm slightly fluctuates.



(a) Case I: $n = 3$ (b) Case II: $n = 3$



(c) Case I: $n = 4$ * (d) Case II: $n = 4$ *

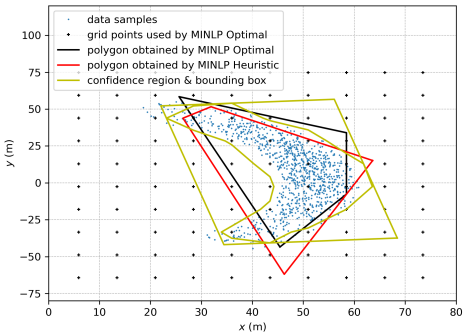


(e) Case I: $n = 5$ (f) Case II: $n = 5$

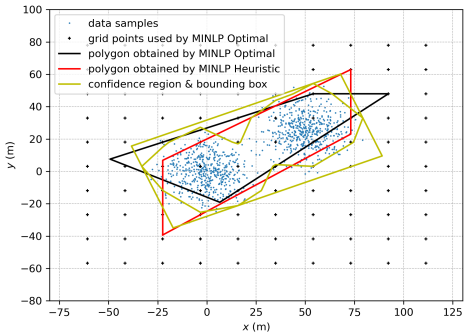
Fig. 15. Comparisons of different numbers of sides n (For Bounding Box $n \equiv 4$)

TABLE I
COMPARISONS OF DIFFERENT NUMBERS OF SIDES n (FOR BOUNDING BOX $n \equiv 4$)

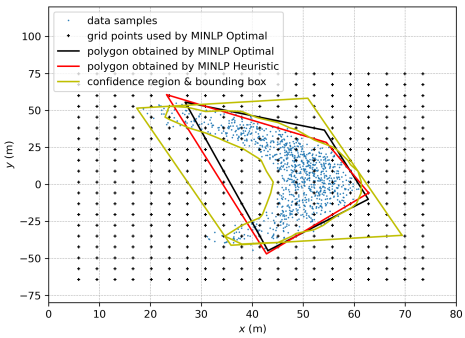
# Sides n	Algorithm	Case I			Case II		
		Ratio	**Area** (m^2)	**Time** (s)	**Ratio**	**Area** (m^2)	**Time** (s)
3 (Triangle)	MINLP Optimal	90.1%	1835.9	27.5	89.1%	4220.7	13.7
	MINLP Heuristic	90.2%	1911.9	0.533	90.1%	4706.6	0.209
	Bounding Box	N/A	N/A	N/A	N/A	N/A	N/A
4 (Quadrilateral) *	MINLP Optimal	91.5%	1827.0	70.5	91.4%	3570.5	18.6
	MINLP Heuristic	90.7%	1807.9	0.801	90.6%	3457.1	0.361
	Bounding Box	97.9%	3235.1	0.216	97.7%	4987.2	0.189
5 (Pentagon)	MINLP Optimal	89.7%	1667.9	316.3	91.3%	3511.6	27.1
	MINLP Heuristic	90.2%	1721.9	1.145	90.5%	3401.7	0.525
	Bounding Box	N/A	N/A	N/A	N/A	N/A	N/A



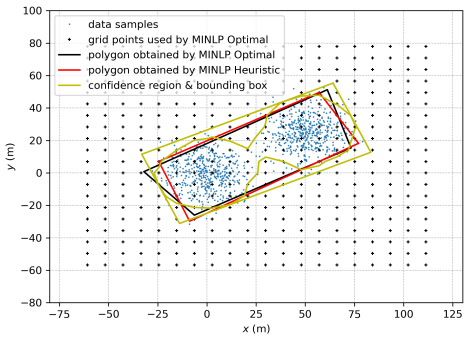
(a) Case I: $N^2 = 10^2$



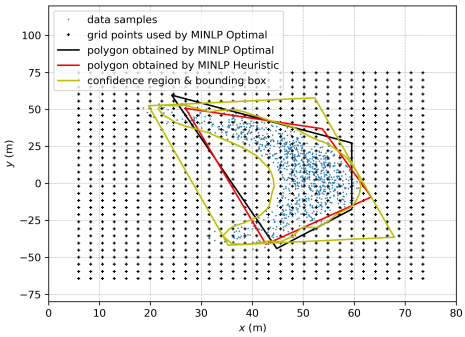
(b) Case II: $N^2 = 10^2$



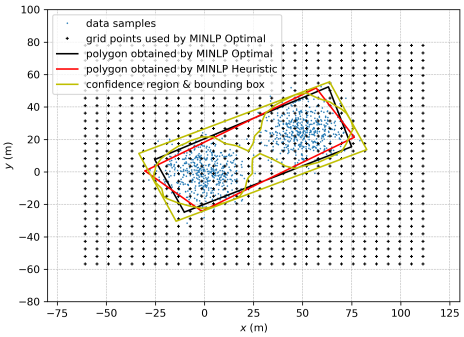
(c) Case I: $N^2 = 20^2$ *



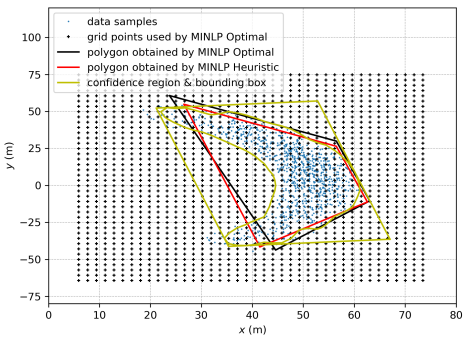
(d) Case II: $N^2 = 20^2$ *



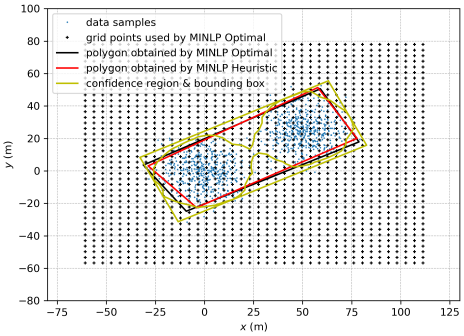
(e) Case I: # $N^2 = 30^2$



(f) Case II: $N^2 = 30^2$



(g) Case I: $N^2 = 40^2$

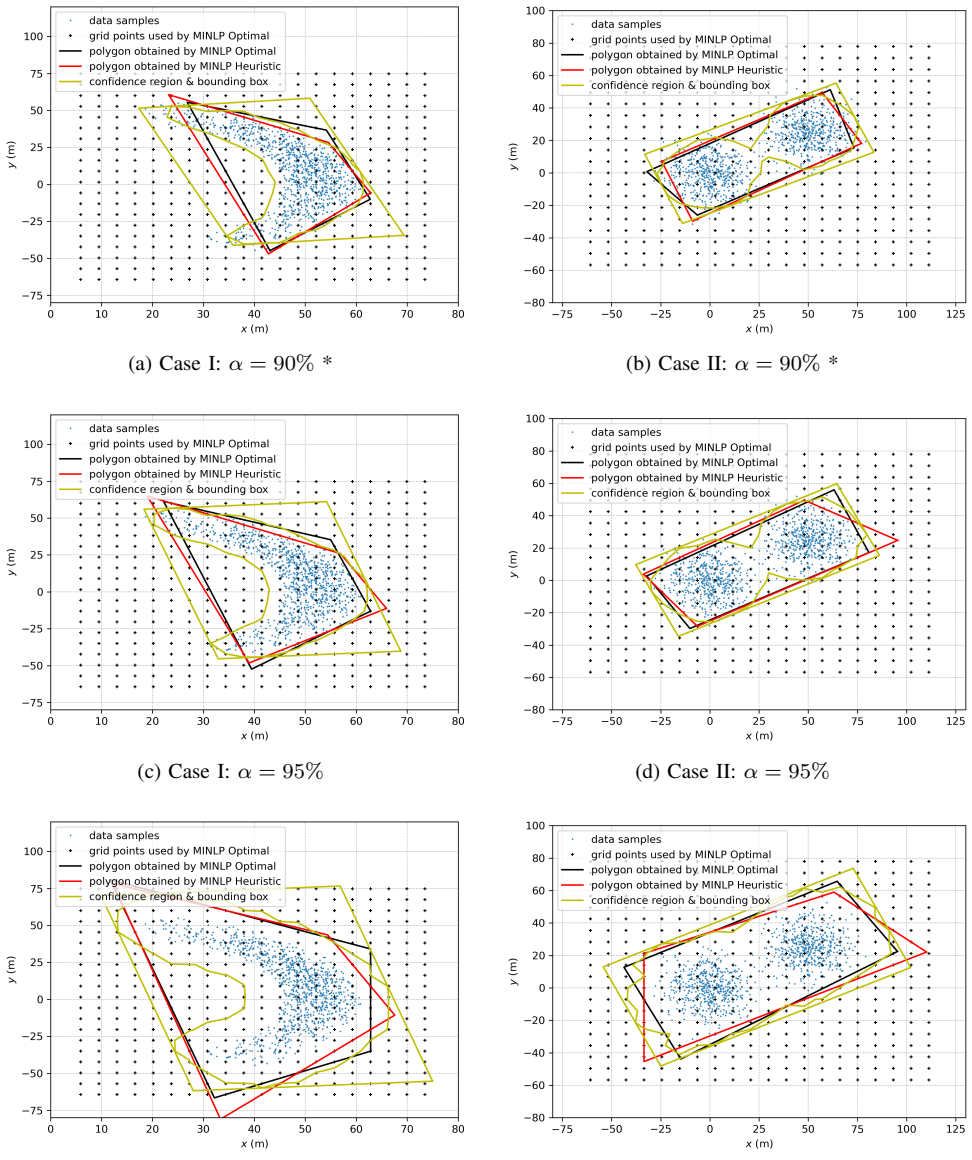


(h) Case II: # $N^2 = 40^2$

Fig. 16. Comparisons of different numbers of gird points N^2

TABLE II
COMPARISONS OF DIFFERENT NUMBERS OF GRID POINTS

# Grid Points N^2	Algorithm	Case I			Case II		
		Ratio	Area (m^2)	Time (s)	Ratio	Area (m^2)	Time (s)
10^2	MINLP Optimal	87.2%	1703.6	0.321	82.2%	3784.7	0.082
	MINLP Heuristic	90.2%	1908.7	0.064	92.6%	4110.5	0.015
	Bounding Box	97.9%	3257.2	0.209	99.6%	6318.8	0.184
20^2 *	MINLP Optimal	91.5%	1872.0	70.5	91.4%	3570.5	18.6
	MINLP Heuristic	90.7%	1807.9	0.801	90.6%	3457.1	0.361
	Bounding Box	97.9%	3235.1	0.216	97.7%	4987.2	0.189
30^2	MINLP Optimal	91.4%	1829.1	1150.2	92.0%	3578.3	362.9
	MINLP Heuristic	90.4%	1755.9	8.9	91.8%	3443.1	6.9
	Bounding Box	97.7%	3157.7	0.225	97.6%	4893.5	0.193
40^2	MINLP Optimal	91.3%	1827.0	2360.1	91.4%	3554.8	1237.4
	MINLP Heuristic	90.1%	1721.1	125.5	90.9%	3425.0	23.8
	Bounding Box	97.1%	3030.2	0.239	97.3%	4722.6	0.211



(a) Case I: $\alpha = 90\%$ *

(b) Case II: $\alpha = 90\%$ *

(c) Case I: $\alpha = 95\%$

(d) Case II: $\alpha = 95\%$

(e) Case I: $\alpha = 99\%$

(f) Case II: $\alpha = 99\%$

Fig. 17. Comparisons of different confidence levels α

TABLE III
COMPARISONS OF DIFFERENT CONFIDENCE LEVELS

Confidence Level α	Algorithm	Case I			Case II		
		Ratio	Area (m^2)	Time (s)	Ratio	Area (m^2)	Time (s)
90% *	MINLP Optimal	91.5%	1872.0	70.5	91.4%	3570.5	18.6
	MINLP Heuristic	90.7%	1807.9	0.801	90.6%	3457.1	0.361
	Bounding Box	97.9%	3235.1	0.216	97.7%	4987.2	0.189
95%	MINLP Optimal	95.7%	2359.6	53.5	96.1%	4312.6	8.8
	MINLP Heuristic	96.1%	2424.7	0.417	96.2%	4471.5	0.112
	Bounding Box	99.4%	3709.1	0.243	99.0%	5613.5	0.235
99%	MINLP Optimal	99.9%	4350.6	50.6	99.9%	7223.7	7.8
	MINLP Heuristic	99.9%	4443.5	0.377	99.9%	7460.1	0.090
	Bounding Box	100%	6314.8	0.303	100%	9539.5	0.271

When α is fixed, the area of the polygon obtained by MINLP Optimal is far less than that of the Bounding Box. This means that the polygon obtained by MINLP Optimal is far less conservative than Bounding Box. However, the computational time of implementing MINLP Optimal is much longer than Bounding Box. That is, MINLP Optimal is inefficient to perform the online evaluation. Instead, the area of the polygon obtained by MINLP Heuristic is close to MINLP Optimal while the computational time of implementing MINLP Heuristic is short enough. Hence, MINLP Heuristic runs much faster than MINLP Optimal while ensuring accuracy.

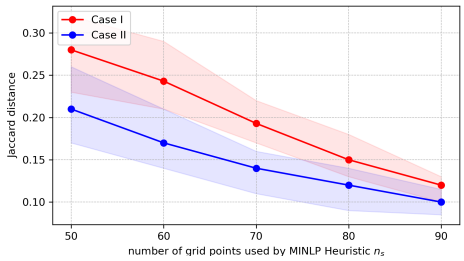
E. Robustness of the MINLP Heuristic Algorithm

As indicated in Algorithm 3, every time we run MINLP Heuristic, we apply a weighted sampling to select representative grid points $\boldsymbol{g}'$ out of the grid points $\boldsymbol{g}$. Due to the randomness of weighted sampling, the selected grid points $\boldsymbol{g}'$ typically differ every time, and the polygon obtained by MINLP Heuristic changes every time it is applied. This leads to the following concerns regarding the robustness of MINLP Heuristic. 1) Space: this refers to whether the convex polygon obtained significantly differs every time; 2) Time: this refers to whether the computational time spent significantly differs every time. A major difference would mean that MINLP Heuristic is not robust enough for applications in real scenarios.

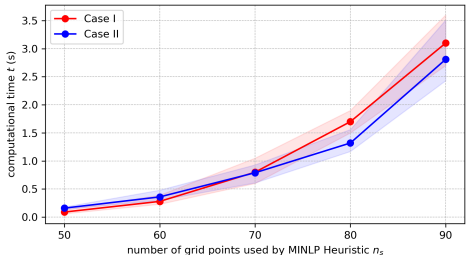
The similarity between two polygons can be quantified by means of the *Jaccard distance*. Formally, for any two sets A and B, Jaccard distance is defined as $d(A,B) = 1 - \frac{\mathrm{Card}(A \cap B)}{\mathrm{Card}(A \cup B)}$ [33], where the operation Card finds the cardinality of a set. A smaller Jaccard distance means the two sets are more similar to each other.

We analyze the robustness of MINLP Heuristic with respect to the varying number N_s of grid points $\boldsymbol{g}'$ used by MINLP Heuristic for Cases I and II mentioned above. For either Case I or Case II, all the parameters, except for the number N_s, are constant and the same as in the respective baselines.

For either Case I or Case II, given a fixed number N_s, we run MINLP Optimal once to obtain a constant optimal polygon and run MINLP Heuristic ten times to obtain ten approximate polygons which may differ every time. We evaluate the Jaccard distance between every approximate polygon and the optimal polygon and obtain ten distances eventually. In addition, we evaluate the computational time of running



(a) Space robustness



(b) Time robustness

Fig. 18. Robustness analysis of MINLP Heuristic algorithm

MINLP Heuristic every time. We repeat the operation above for $N_s = 50, 60, 70, 80, 90$, respectively. The results are depicted in Fig. 18. As illustrated in Fig. 18a, for both cases, when N_s is too small, like $N_s = 50$, different Jaccard distances fluctuate significantly around the average value, suggesting that MINLP Heuristic has poor robustness under this circumstance. As N_s increases from 50 to 90, the average Jaccard distance decreases from 0.28, 0.21 to 0.12, 0.10, respectively. This means the approximate polygon obtained by MINLP Heuristic becomes more near-optimal with respect to an increasing N_s. Moreover, the variance of ten Jaccard distances also decreases as N_s grows. That is, MINLP Heuristic becomes more robust with respect to the increase of N_s. As suggested in Fig. 18b, as N_s increases, the average computational time increases from 0.09 s to 3.1 s for Case I, and from 0.16 s to 2.81 s for Case II. Additionally, as N_s grows, the variance of computational time increases.

In summary, the robustness of MINLP Heuristic comes at the cost of losing computational efficiency. For the case studies under consideration, a balance is achieved for $N_s = 70$ for Case I or $N_s = 60$ for Case II. This makes the MINLP Heuristic

robust enough and its applications promising.

VII. Conclusion

This paper proposes an efficient convex approximation of the PRS of an uncertain dynamic system. For arbitrary unknown uncertainties, especially unbounded ones, the notion of the PRS is introduced as an extension of the traditional reachable set, connecting chance constraint formulation and reachability analysis. In this paper, the PRS is obtained through a data-driven approach employing a KDE method. FFT is then customized to accelerate the computation of KDE to make it computationally attractive. The irregularity or non-convexity of the PRS refrains its use in optimal design applications. To address this issue, we formulate a MINLP problem whose solution accounts for a convex approximation of the PRS. We then develop a MINLP Heuristic algorithm to solve it. Comprehensive case studies demonstrate that our proposed algorithm enjoys the benefits of accuracy, efficiency, near-optimality, as well as robustness while providing a convex approximation for the PRS.

This work is limited to a 2D system. Future work will be devoted to solving higher dimensional problems, and applying the results to the design of safety-critical real-time motion planning algorithms for uncertain robotic systems.

VIII. Appendix

A. Proof of Lemma 1

I) First, we prove that any two formally different representations of lines refer to different lines.

Given $n \geq 3$ planar lines not passing through the origin $l^k := a_k x + b_k y - 1 = 0, k \in \{1, \ldots, n\}$, we can construct a sequence of lines $(l^1, l^2, \ldots, l^n, l^1)$. Two formally different lines l^i, l^j in the sequence are defined to be consecutive if and only if $j = i^{\oplus} \vee j = i^{\ominus}$. Since there are n intersection points $V_{ij} := l^i \cap l^j, i \in \{1, \ldots, n\}, j = i^{\oplus}$, it follows that $l^i \neq l^j,\ i \in \{1, \ldots, n\}, j = i^{\oplus}$. Thus, any two formally different lines consecutive in the sequence are indeed different from each other.

For $n = 3$, the sequence of lines is (l^1, l^2, l^3, l^1). Since any two formally different lines in the sequence (l^1, l^2, l^3, l^1) are consecutive, thus l^1, l^2, l^3 are indeed different from each other. For $n \geq 4$, there are at least one pair of formally different lines not consecutive in the sequence. We have already shown that any two formally different lines consecutive in the sequence are indeed different from each other, and therefore it remains to show that any two formally different lines not consecutive in the sequence are also indeed different from each other, given the conditions of this lemma. We prove this by contradiction. Suppose that $l^u, l^v, u \in \{1, \ldots, n\}, v \in \{1, \ldots, n\} - \{u^{\ominus}, u, u^{\oplus}\}$ are two formally different lines not consecutive in the sequence but are indeed an identical line. Then $V_{v^{\ominus}v}$ and $V_{vv^{\oplus}}$ are both on the line l^u, which means

$$\begin{aligned} &- a_u \left(b_{v^{\ominus}} - b_v\right) + b_u \left(a_{v^{\ominus}} - a_v\right) - \left(a_{v^{\ominus}} b_v - b_{v^{\ominus}} a_v\right) = 0, \\ &- a_u \left(b_v - b_{v^{\oplus}}\right) + b_u \left(a_v - a_{v^{\oplus}}\right) - \left(a_v b_{v^{\oplus}} - b_v a_{v^{\oplus}}\right) = 0. \end{aligned} \tag{20}$$

However, since $v \notin \{u^{\ominus}, u, u^{\oplus}\}$, it follows that the subscript of $V_{v^{\ominus}v}$ is formally different from $V_{u^{\ominus}u}$ and $V_{uu^{\oplus}}$, and the subscript of $V_{vv^{\oplus}}$ is also formally different from $V_{u^{\ominus}u}$ and $V_{uu^{\oplus}}$. Thus, since the constraints Eq. (10) are satisfied, we obtain two instances

$$\begin{aligned} &- a_u \left(b_{v^{\ominus}} - b_v\right) + b_u \left(a_{v^{\ominus}} - a_v\right) - \left(a_{v^{\ominus}} b_v - b_{v^{\ominus}} a_v\right) < 0, \\ &- a_u \left(b_v - b_{v^{\oplus}}\right) + b_u \left(a_v - a_{v^{\oplus}}\right) - \left(a_v b_{v^{\oplus}} - b_v a_{v^{\oplus}}\right) < 0, \end{aligned}$$

which contradicts Eq. (20).

Thus, if constraints Eq. (9) and Eq. (10) are satisfied, any two formally different lines are indeed different from each other.

II) Next, we show that if there are n intersection points $V_{ij} := l^i \cap l^j, i \in \{1, \ldots, n\}, j = i^{\oplus}$ (namely, $D_{ij} := a_i b_j - b_i a_j > 0,\ i \in \{1, \ldots, n\}, j = i^{\oplus}$ according to Eq. (9)), and the constraints Eq. (10) are satisfied, then any two formally different intersection points are indeed different from each other.

We can prove this by contradiction. Assume that there are two formally different intersection points $V_{ij}, V_{uv}, i \in \{1, \ldots, n\}, j = i^{\oplus}, u \in \{1, \ldots, n\} - \{i\}, v = u^{\oplus}$ which are indeed an identical intersection point. Since $n \geq 3$, it follows that $(u \neq i \wedge u \neq j) \vee (v \neq i \wedge v \neq j)$, and without loss of generality, we assume $v \neq i \wedge v \neq j$. Thus, the constraints Eq. (10) apply to V_{ij} and l^v, which yields an instance

$$-a_v \left(b_i - b_j\right) + b_v \left(a_i - a_j\right) - \left(a_i b_j - b_i a_j\right) < 0. \tag{21}$$

However, since V_{ij}, V_{uv} are indeed an identical intersection point, V_{ij} is on the line l^v, meaning that

$$-a_v \left(b_i - b_j\right) + b_v \left(a_i - a_j\right) - \left(a_i b_j - b_i a_j\right) = 0,$$

which contradicts Eq. (21).

Combining I) and II), we conclude that if constraints Eq. (9) and Eq. (10) are satisfied, any two formally different representations of intersection points (resp. lines) refer to different intersection points (resp. lines), and therefore all n intersection points (resp. lines) are indeed different from each other.

B. Proof of Theorem 1

I) Since there are $n \geq 3$ intersection points $V_{ij} := l^i \cap l^j,\ i \in \{1, \ldots, n\}, j = i^{\oplus}$, we can enumerate them in a sequence of intersection points $S_g := (V_{12}, V_{23}, \ldots, V_{(n-1)n}, V_{n1}, V_{12})$. According to Lemma 1, since the coefficients of these lines satisfy constraints Eq. (9) and Eq. (10), then any two formally different intersection points (resp. lines) are indeed different from each other.

II) We can also construct a finite sequence of symbols $S_s := (\text{“}V_{12}\text{”}, \text{“}V_{23}\text{”}, \ldots, \text{“}V_{(n-1)n}\text{”}, \text{“}V_{n1}\text{”}, \text{“}V_{12}\text{”})$. In S_s, 1) since $n \geq 3$, there are at least three formally different symbols “V_{12}”, “V_{23}” and “V_{31}”; 2) No symbols appear more than once except for the case in which only the first symbol “V_{12}” and last symbol “V_{12}” are formally identical. Therefore, Definition 9 applies and S_s is formally enclosing.

Consider a map f_{tag} from S_s to S_g given by the operation of labelling the intersection points, i.e., $V_{ij} := l^i \cap l^j,\ i \in \{1, \ldots, n\}, j = i^{\oplus}$. It can be verified that f_{tag} is injective, surjective and order-preserving. The map f_{tag} also satisfies Condition 4) in Definition 10. This is because in S_s, for any

two consecutive symbols "$V_{i^{\ominus}i}$" and "$V_{ii^{\oplus}}$", and any third symbol "V_{uv}" which is formally different from each of "$V_{i^{\ominus}i}$" and "$V_{ii^{\oplus}}$", according to Eq. (10) we obtain

$$\begin{aligned}&-a_i\left(b_u-b_v\right)+b_i\left(a_u-a_v\right)-\left(a_ub_v-b_ua_v\right)\neq 0,\\&u\notin\{i^{\ominus},i\},v=u^{\oplus},\end{aligned}$$

which means that the intersection point V_{uv} is not on the line l^i. According to I), since formally different symbols "$V_{i^{\ominus}i}$" and "$V_{ii^{\oplus}}$" refer to different intersection points $V_{i^{\ominus}i}$ and $V_{ii^{\oplus}}$, thus these two intersection points determine a non-degenerate line segment. Further, the line segment determined by these two points is a segment of the line l^i. Hence, the intersection point V_{uv} that the third symbol "V_{uv}" refers to is not on the edge $(V_{i^{\ominus}i},V_{ii^{\oplus}})$. In summary, the map f_{tag} satisfies all the conditions in Definition 10 and therefore is an isomorphism type A.

According to Remark 1, since S_s is formally enclosing, and f_{tag} is an isomorphism type A, it follows that the graph G formed by S_g is enclosing.

III) Since $n\geq 3$, the number of formally different symbols in S_s is $n\geq 3$.

As concluded in II), the map f_{tag} is injective, surjective, and order-preserving. The map f_{tag} also satisfies Condition 4) in Definition 12. This is because, for any three consecutive symbols "$V_{i^{\ominus}i}$", "$V_{ii^{\oplus}}$", "$V_{i^{\oplus}i^{\oplus\oplus}}$" in S_s, applying Eq. (10) yields

$$-a_i\left(b_{i^{\oplus}}-b_{i^{\oplus\oplus}}\right)+b_i\left(a_{i^{\oplus}}-a_{i^{\oplus\oplus}}\right)-\left(a_{i^{\oplus}}b_{i^{\oplus\oplus}}-b_{i^{\oplus}}a_{i^{\oplus\oplus}}\right)\neq 0,$$

which means that the intersection point $V_{i^{\oplus}i^{\oplus\oplus}}$ is not on the line l^i. From I), since formally different symbols "$V_{i^{\ominus}i}$" and "$V_{ii^{\oplus}}$" refer to different intersection points $V_{i^{\ominus}i}$ and $V_{ii^{\oplus}}$, then $V_{i^{\ominus}i}$ and $V_{ii^{\oplus}}$ are different from each other. Further, the intersection point $V_{i^{\ominus}i}$ and $V_{ii^{\oplus}}$ are on the line l^i. Hence, the three intersection points $V_{i^{\ominus}i}$, $V_{ii^{\oplus}}$, $V_{i^{\oplus}i^{\oplus\oplus}}$ are not collinear. In summary, the map f_{tag} satisfies all the conditions in Definition 12 and therefore is an isomorphism type B.

According to Remark 2, since f_{tag} is an isomorphism type B, and there are $n\geq 3$ formally different symbols in S_s, it follows that the graph G formed by S_g is non-degenerate.

IV) We have shown that the graph G formed by S_g is an enclosing n sided polygon. For a line $l^k,k\in\{1,\ldots,n\}$, once the constraints Eq. (10) are satisfied, it follows that

$$\begin{cases}-a_k\left(b_i-b_j\right)+b_k\left(a_i-a_j\right)-\left(a_ib_j-b_ia_j\right)<0,\\ i\in\{1,\ldots,n\}-\{k^{\ominus},k\},j=i^{\oplus},\\ \\ -a_k\left(b_i-b_j\right)+b_k\left(a_i-a_j\right)-\left(a_ib_j-b_ia_j\right)=0,\\ i\in\{k^{\ominus},k\},j=i^{\oplus}.\end{cases}\tag{22}$$

Since there are n intersection points $V_{ij},\ i\in\{1,\ldots,n\},j=i^{\oplus}$, according to Eq. (9), we have $D_{ij}:=a_ib_j-b_ia_j>0,\ i\in\{1,\ldots,n\},j=i^{\oplus}$. Thus, the inequalities Eq. (22) can be transformed to

$$\begin{cases}a_k\frac{-(b_i-b_j)}{D_{ij}}+b_k\frac{(a_i-a_j)}{D_{ij}}-1<0,\\ i\in\{1,\ldots,n\}-\{k^{\ominus},k\},\ j=i^{\oplus},\\ \\ a_k\frac{-(b_i-b_j)}{D_{ij}}+b_k\frac{(a_i-a_j)}{D_{ij}}-1=0,\\ i\in\{k^{\ominus},k\},\ j=i^{\oplus},\end{cases}\tag{23}$$

which means that all n intersection points $V_{ij},i\in\{1,\ldots,n\},j=i^{\oplus}$ are on the same side of the line l^k. Therefore, in S_g, for any two consecutive intersection points V_{ij},V_{uv} where $i\in\{1,\ldots,n\}-\{k^{\ominus}\},j=i^{\oplus},u=i^{\oplus},v=i^{\oplus\oplus}$, we can define a set constructed by the convex combination of V_{ij} and V_{uv} excluding these two boundary points, which is

$$\begin{aligned}&\Big\{t(-\frac{b_i-b_j}{D_{ij}},\frac{a_i-a_j}{D_{ij}})+(1-t)(-\frac{b_u-b_v}{D_{uv}},\frac{a_u-a_v}{D_{uv}}):\\&\ t\in\mathbb{R}\wedge 0<t<1\Big\}.\end{aligned}$$

It is a subset of the half plane $\{(x,y):a_kx+b_ky-1<0\}$. In other words, the edge $(V_{ii^{\oplus}},V_{i^{\oplus}i^{\oplus\oplus}}),i\in\{1,\ldots,n\}-\{k^{\ominus}\}$, is on the strict inner side of the line l^k. This is because

$$\begin{aligned}&a_k\left(t\frac{-(b_i-b_j)}{D_{ij}}+(1-t)\frac{-(b_u-b_v)}{D_{uv}}\right)+\\&b_k\left(t\frac{(a_i-a_j)}{D_{ij}}+(1-t)\frac{(a_u-a_v)}{D_{uv}}\right)-1\\&=t\left(a_k\frac{-(b_i-b_j)}{D_{ij}}+b_k\frac{(a_i-a_j)}{D_{ij}}\right)+\\&(1-t)\left(a_k\frac{-(b_u-b_v)}{D_{uv}}+b_k\frac{(a_u-a_v)}{D_{uv}}\right)-1\\&<0,\end{aligned}\tag{24}$$

according to Eq. (23). For the polygon G formed by S_g, according to Eq. (24), we have shown that every edge $(V_{ii^{\oplus}},V_{i^{\oplus}i^{\oplus\oplus}}),i\in\{1,\ldots,n\}-\{k^{\ominus}\}$ of the polygon is on the same side of the line $l^k,k\in\{1,\ldots,n\}$ that the edge $(V_{k^{\ominus}k},V_{kk^{\oplus}}),k\in\{1,\ldots,n\}$ defines. Recall the definition of a convex polygon: For each edge of the polygon, all the other edges are on the same side of the line that the edge defines. Therefore, the polygon G formed by S_g is a convex polygon.

V) Recall that for an arbitrary point (u,v) and an enclosing n sided convex polygon determined by n lines $l^k:=a_kx+b_ky-1=0,k\in\{1,\ldots,n\}$, if $\bigwedge_{k=1}^{n}a_ku+b_kv-1<0$ holds, then the point (u,v) lies inside the polygon. We have shown above that if constraints Eq. (9) and Eq. (10) are satisfied, then the graph G formed by S_g is an enclosing n sided convex polygon. Since $a_k\cdot 0+b_k\cdot 0-1\equiv -1<0,k\in\{1,\ldots,n\}$, the origin $(0,0)$ lies inside the polygon G formed by S_g.

From the above, we conclude that if constraints Eq. (9) and Eq. (10) are satisfied, then the graph G formed by S_g is an enclosing n sided convex polygon with the origin $(0,0)$ inside. For such $n\geq 3$ planar lines $l^k:=a_kx+b_ky-1=0,\ k\in\{1,\ldots,n\}$ not passing through the origin satisfying constraints Eq. (9) and Eq. (10), the boundary of the region S generated by these lines, as defined in Eq. (8), is exactly the graph G formed by S_g. Since that graph G is an enclosing n

sided convex polygon with the origin in its interior under the conditions of Eq. (9) and Eq. (10), it follows that the boundary of the region S generated by these lines is an enclosing n sided convex polygon with the origin in its interior.

C. Proof of Corollary 1

This follows from the application of an affine coordinate transformation using $(\bar{x}, \bar{y})$. More precisely, given coordinates x-y, consider the transformation $(x^{\text{new}}, y^{\text{new}}) = (x - \bar{x}, y - \bar{y})$. In the new coordinates, the origin $(0^{\text{new}}, 0^{\text{new}})$ is located at $(\bar{x}, \bar{y})$. In addition, the representation of lines changes to

$$\left\{a(x - \bar{x}) + b(y - \bar{y}) - 1 = 0 : a, b \in \mathbb{R} \wedge a^2 + b^2 \neq 0\right\}.$$

From Theorem 1 it follows that Eq. (9) and Eq. (10) are sufficient to guarantee that the boundary of the region $S := \left\{(x^{\text{new}}, y^{\text{new}}) : \bigwedge_{k=1}^{n} a_k x^{\text{new}} + b_k y^{\text{new}} - 1 \leq 0\right\}$ is an enclosing n sided convex polygon with the origin $(0^{\text{new}}, 0^{\text{new}})$ in its interior. By changing back to the original coordinate system, the result follows.

Acknowledgment

This work was supported by the National Science Foundation under Grants CMMI-2138612. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not reflect the views of NSF. The authors would like to thank Lin Li, Paul Lathrop from UC San Diego and Jun Xiang from San Diego State University for their valuable comments and discussions.

References

[1] T. Lew, A. Sharma, J. Harrison, A. Bylard, and M. Pavone, "Safe active dynamics learning and control: A sequential exploration–exploitation framework," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2888–2907, 2022.

[2] Z. Xing, H. Yue, T. Zhang, W. Wu, and R. Hu, "Collision avoidance and give way of heterogeneous and variable-sized multiple mobile robots based on glued nodes," *IEEE Transactions on Automation Science and Engineering*, 2023.

[3] Y. Wen and P. Pagilla, "Path-constrained and collision-free optimal trajectory planning for robot manipulators," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 763–774, 2022.

[4] P. Wu, J. Xie, Y. Liu, and J. Chen, "Risk-bounded and fairness-aware path planning for urban air mobility operations under uncertainty," *Aerospace Science and Technology*, vol. 127, p. 107738, 2022.

[5] A. Paudel, S. Gupta, M. Thapa, S. B. Mulani, and R. W. Walters, "Higher-order taylor series expansion for uncertainty quantification with efficient local sensitivity," *Aerospace Science and Technology*, vol. 126, p. 107574, 2022.

[6] P. Wu, X. Yang, P. Wei, and J. Chen, "Safety assured online guidance with airborne separation for urban air mobility operations in uncertain environments," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[7] F. Meng, L. Chen, H. Ma, J. Wang, and M. Q.-H. Meng, "Learning-based risk-bounded path planning under environmental uncertainty," *IEEE Transactions on Automation Science and Engineering*, 2023.

[8] L. Zhang, X. Zhang, X. Chang, and N. Zhao, "Adaptive fault-tolerant control-based real-time reachable set synthesis of heterogeneous nonlinear singular multiagent systems with uncertain parameters," *IEEE Transactions on Automation Science and Engineering*, 2023.

[9] Z. Li, "Comparison between safety methods control barrier function vs. reachability analysis," *arXiv preprint arXiv:2106.13176*, 2021.

[10] M. Althoff, G. Frehse, and A. Girard, "Set propagation techniques for reachability analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 369–395, 2021.

[11] A. Devonport, F. Yang, L. El Ghaoui, and M. Arcak, "Data-driven reachability analysis with christoffel functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 5067–5072.

[12] W. Dai, B. Pang, and K. H. Low, "Conflict-free four-dimensional path planning for urban air mobility considering airspace occupancy," *Aerospace Science and Technology*, p. 107154, 2021.

[13] W. Han, A. Jasour, and B. Williams, "Non-gaussian risk bounded trajectory optimization for stochastic nonlinear systems in uncertain environments," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 044–11 050.

[14] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin, "Decomposition of reachable sets and tubes for a class of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3675–3688, 2018.

[15] P. Wu and J. Chen, "Online probabilistic collision detection for urban air mobility under data-driven uncertainty," in *AIAA SCITECH 2023 Forum*, 2023, p. 2539.

[16] Y. Zou, H. Zhang, G. Zhong, H. Liu, and D. Feng, "Collision probability estimation for small unmanned aircraft systems," *Reliability Engineering & System Safety*, vol. 213, p. 107619, 2021.

[17] Y. Yang, J. Zhang, K.-Q. Cai, and M. Prandini, "Multi-aircraft conflict detection and resolution based on probabilistic reach sets," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 309–316, 2016.

[18] L. Hewing, K. P. Wabersich, and M. N. Zeilinger, "Recursively feasible stochastic model predictive control using indirect feedback," *Automatica*, vol. 119, p. 109095, 2020.

[19] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[20] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for mavs in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.

[21] P. Lathrop, B. Boardman, and S. Martínez, "Distributionally safe path planning: Wasserstein safe rrt," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 430–437, 2021.

[22] S. Boone and J. McMahon, "Spacecraft maneuver design with non-gaussian chance constraints using gaussian mixtures," in *2022 AAS/AIAA Astrodynamics Specialist Conference*, 2022.

[23] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[24] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[25] V. Lefkopoulos and M. Kamgarpour, "Using uncertainty data in chance-constrained trajectory planning," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2264–2269.

[26] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[27] H. Han, S. Fu, H. Sun, and J. Qiao, "Data-driven multimodel predictive control for multirate sampled-data nonlinear systems," *IEEE Transactions on Automation Science and Engineering*, 2022.

[28] C. Ericson, *Real-time collision detection*. Crc Press, 2004.

[29] M. P. Wand and M. C. Jones, *Kernel smoothing*. CRC press, 1994.

[30] K. Ding and Q. Zhu, "Intermittent static output feedback control for stochastic delayed-switched positive systems with only partially measurable information," *IEEE Transactions on Automatic Control*, 2023.

[31] A. Alanwar, A. Koch, F. Allgöwer, and K. H. Johansson, "Data-driven reachability analysis using matrix zonotopes," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 163–175.

[32] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[33] P. Wu and J. Chen, "Online evaluation for chance-constrained geofences under data-driven uncertainties," in *AIAA AVIATION 2022 Forum*, 2022, p. 3613.

[34] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009, vol. 27.

[35] R. Zach, "Sets, logic, computation: An open introduction to metalogic," 2021.

[36] I. Griva, S. G. Nash, and A. Sofer, *Linear and nonlinear optimization*. Siam, 2009, vol. 108.

[37] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of applied optimization*, vol. 1, no. 1, pp. 65–77, 2002.

[38] P. S. Efraimidis and P. G. Spirakis, "Weighted random sampling with a reservoir," *Information processing letters*, vol. 97, no. 5, pp. 181–185, 2006.