Chance-Constrained Optimal Path Planning With Obstacles

Chance-Constrained Optimal Path Planning with Obstacles

Lars Blackmore, Masahiro Ono and Brian C. Williams

***Abstract*—Autonomous vehicles need to plan trajectories to a specified goal that avoid obstacles. For robust execution, we must take into account uncertainty, which arises due to uncertain localization, modeling errors, and disturbances. Prior work handled the case of set-bounded uncertainty. We present here a chance-constrained approach, which uses instead a probabilistic representation of uncertainty. The new approach plans the future probabilistic distribution of the vehicle state so that the probability of failure is below a specified threshold. Failure occurs when the vehicle collides with an obstacle, or leaves an operator-specified region. The key idea behind the approach is to use bounds on the probability of collision to show that, for linear-Gaussian systems, we can approximate the nonconvex chance-constrained optimization problem as a Disjunctive Convex Program. This can be solved to global optimality using branch-and-bound techniques. In order to improve computation time, we introduce a customized solution method that returns almost-optimal solutions along with a hard bound on the level of suboptimality. We present an empirical validation with an aircraft obstacle avoidance example.**

I. Introduction

Path planning for autonomous vehicles such as Unmanned Air Vehicles (UAVs) has received a great deal of attention in recent years [1][2][3][4]. A UAV needs to be able to plan trajectories that take the aircraft from its current location to a goal, while avoiding obstacles. These trajectories should be optimal with respect to a criterion such as time or fuel consumption. This problem is challenging for two principal reasons. First, the optimization problem is inherently nonconvex due to the presence of obstacles in the feasible space. Second, there are a number of sources of uncertainty in the problem, such as disturbances, uncertain localization and modeling uncertainty.

Previous approaches addressed the first of these challenges. In our work we build on [5], which uses a Mixed-Integer Linear Programming (MILP) approach to design fuel-optimal trajectories for vehicles modeled as linear systems. The Mixed-Integer Linear Programming approach uses highly-optimized commercial software[6] based on branch-and-cut and a host of other techniques to make the non-convex optimization problem tractable[7]. Throughout this paper we similarly assume linear system dynamics. Prior work (for example [5] and [8]) has shown that linear system models can be used to design trajectories for vehicles such as UAVs and satellites.

The MILP approach of [5] does not explicitly take into account uncertainty. That is, it is assumed in [5] that the knowledge of the vehicle's state is perfect, and that the planned path can be executed perfectly. In practice the executed path will deviate from the planned path and can collide with obstacles, even if the planned path did not. For a vehicle such as a UAV, uncertainty arises for three main reasons. First, aircraft location is not usually known exactly, but is estimated using a system model, inertial sensors and/or a Global Positioning System. Second, system models are approximations of the true system model, and the system dynamics themselves are usually not fully known. Third, disturbances act on the aircraft that make the true trajectory deviate from the planned trajectory.

The problem of path planning under uncertainty was previously addressed for the case of set-bounded uncertainty models[9]. In the case of disturbances this corresponds to having a known bound on the magnitude of the disturbance. Robustness is achieved by designing trajectories that guarantee feasibility of the plan as long as disturbances do not exceed these bounds. In the present paper, we use an alternative approach that characterizes uncertainty in a probabilistic manner, and finds the optimal sequence of control inputs subject to the constraint that the probability of failure must be below a user-specified threshold. This constraint is known as a *chance constraint*[10].

In many cases, the probabilistic approach to uncertainty modeling has a number of advantages over a set-bounded approach. Disturbances such as wind are best represented using a stochastic model, rather than a set-bounded one[11]. When using a Kalman Filter for localization, the state estimate is provided as a probabilistic distribution specifying the mean and covariance of the state. In addition, by specifying the probability that a plan is executed successfully, the operator can stipulate the desired level of conservatism in the plan in a meaningful manner, and can trade conservatism against performance.

A great deal of work has taken place in recent years relating to chance-constrained optimal control of linear systems subject to Gaussian uncertainty in convex regions[12], [13], [14], [15], [16], [17], [18]. In the present paper we extend this to the problem of chance-constrained path planning with obstacles, i.e. in nonconvex regions. The key idea is to bound the probability of collision with obstacles to give a conservative approximation of the full chance-constrained problem. We provide a new bound that approximates the chance-constrained problem as a Disjunctive Convex Program. This can be solved to global optimality using branch-and-bound techniques. In order to make the computational complexity practical for onboard use, we introduce a customized solution method for the Disjunctive Convex Program. This makes use of two new disjunctive linear bounds on the probability of colli-

L. Blackmore is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109.

M. Ono and B. C. Williams are with the Massachusetts Institute of Technology, Cambridge, MA 02139.

sion, one of which provides a lower bound on the optimal cost of the Disjunctive Convex Program, while the second provides an upper bound on the optimal cost. Using these bounds with highly optimized software for Disjunctive Linear Programming[6], the customized solution approach is able to reduce the solution time required dramatically and, in almost all cases, provide tight bounds on the suboptimality introduced by not solving the full Disjunctive Convex Program. We demonstrate our approach in simulation and show that the conservatism introduced by the approach is small.

II. Related Work

A large body of work exists on the topic of path planning with deterministic system models, including [1], [2], [3], [4], [19]. We do not intend to provide a review of this field, but instead refer the reader to the review [20] and the books [21] and [22]. Of this work, in the present paper we extend [5], which introduced a Mixed-Integer Linear Programming approach that designs fuel-optimal trajectories for vehicles modeled as linear systems. The MILP approach has the advantage that the resulting optimization problem can be solved to global optimality using efficient commercial solvers[6]. Refs. [8] and [3] extended this approach to solve problems in aircraft and spacecraft trajectory planning. By including temporally flexible state plans, [23] was able to generate optimal trajectories for UAVs with time-critical mission plans.

Optimal planning under set-bounded uncertainty has received a great deal of attention in the robust Model Predictive Control (MPC) community[24], [9], [25], [26], [27]. We refer the interested reader to the review [28] and the references therein for a more extensive survey. The majority of work in the MPC literature assumes a convex feasible region. However the case of obstacle avoidance, i.e. nonconvex feasible regions, was handled in [9]. In recent years the MPC community has generated a number of results relating to chance-constrained optimal control of linear systems in convex regions. The problem of designing optimal feedforward control sequences for a fixed feedback structure was considered by [12], [14], [15], [17], [18], [29]. This work was extended to the problem of coupled feedforward and feedback control by [30], [31], [32]. This extension enables the variance of the future state distribution to be optimized, which is not the case with just feedforward control. Non-Gaussian uncertainty was handled by [16], [18], [33] using sampling approaches.

For chance-constrained planning in nonconvex feasible regions, early results in the literature suggested simply converting the problem into a set-bounded one, by ensuring that the 3-sigma confidence region does not collide with obstacles[34], [35], [36]. Once this is done, standard approaches for set-bounded uncertainty can be employed. This approach has been shown to be conservative by orders of magnitude, meaning that in many cases, the approach will fail to find a feasible solution even if one exists[18]. To the authors' knowledge, the only prior literature to handle nonconvex chance-constrained planning without resorting to over-conservative set-conversion techniques is the authors' own work[37], [38], [39], [40], [33]. The approach in [39], [40], [33] uses samples, or 'particles', to approximate the planning problem. The particle-based approach approximates the chance constraint, and hence does not guarantee satisfaction of the constraint. By contrast, the approach in the present paper uses an analytic bound to ensure satisfaction of the constraint. In addition, while the particle-based approach applies to arbitrary uncertainty distributions, rather than only Gaussian distributions, it is significantly more computationally intensive than the bounding approach proposed in the present paper. In Section XI-C we provide an empirical comparison of the new approach and the approach of [33], which demonstrates these points. These results also show that the new approach introduces very little conservatism, unlike the set conversion techniques proposed by [34], [35], [36].

Early forms of the results in this paper were presented in [37], [38]. The present paper extends these results by unifying the bounds presented in [37], [38] and deriving a new suboptimal solution algorithm that reduces computation time while providing, in almost all cases, tight bounds on the suboptimality introduced.

III. Problem Statement

In this work, we consider the case where there is uncertainty in the problem that can be described probabilistically. We consider three sources of uncertainty:

1) The initial position of the vehicle is specified as a probabilistic distribution over possible positions. This distribution would typically be generated with an estimation technique such as Kalman filtering, using noisy measurements from inertial sensors and global positioning data. In this work we assume that the initial position of the vehicle is specified as a Gaussian distribution.
2) Disturbances act on the vehicle. These are modeled as a Gaussian noise process added to the system dynamics. In the case of an aircraft, this process represents accelerations caused by wind.
3) The system model is not known exactly. Uncertainty in the system model may arise due to modeling errors or linearization. We assume that model uncertainty can be modeled as a Gaussian white noise process added to the system dynamic equations [41].

Throughout this paper we assume a linear, discrete-time system model. Linear system dynamics are valid for vehicles such as satellites operating close to a reference orbit, or a UAV operating with an inner-loop feedback controller. In the latter, the key idea is that, while the low-level dynamics of the system are nonlinear, the controlled plant from reference position to true position can be approximated as a low-order, linear system for the purposes of path planning[5].

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \omega_t. \tag{1}$$

Here, $\mathbf{x}_t$ is the system state at time step t and $\mathbf{u}_t$ is the control input at time step t. The variable ω_t is a Gaussian white noise process that represents disturbances and model uncertainty, and is distributed according to $\omega_t \sim \mathcal{N}(0, Q)$. The assumption of zero mean, white noise is made to simplify the notation; the methods described in this paper apply equally to

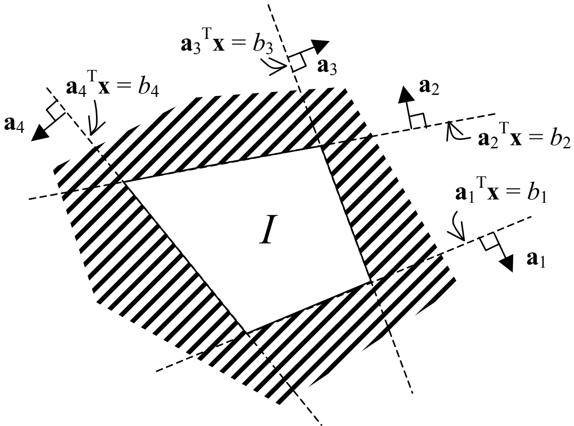

Fig. 1. Polyhedral stay-in region $\mathcal{I}$ encoded as a conjunction of linear inequality constraints. The vehicle must remain in the unshaded region.

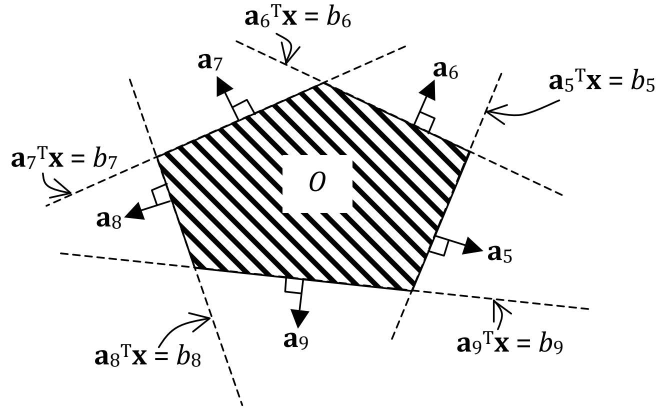

Fig. 2. Polyhedral obstacle $\mathcal{O}$ encoded as a disjunction of linear equality constraints. The vehicle must avoid the shaded region.

Gaussian colored, non-zero-mean noise as long as the statistics are known.

In this paper we consider polygonal convex *stay-in* regions in which the system state must remain, and polygonal convex obstacles, outside of which the system state must remain[1]. A stay-in region $\mathcal{I}$ is defined as a conjunction of linear constraints as follows:

$$\mathcal{I} \Longleftrightarrow \bigwedge_{t \in T(\mathcal{I})} \bigwedge_{i \in G(\mathcal{I})} \mathbf{a}_i' \mathbf{x}_t \leq b_i, \tag{2}$$

where $G(\mathcal{I})$ is a set containing the indices of the linear constraints defining the region, and $T(\mathcal{I})$ is the set of time steps at which the stay-in region applies. We use $\mathbf{v}'$ to denote the transpose of vector $\mathbf{v}$. An obstacle $\mathcal{O}$ is defined as a disjunction of linear constraints as follows:

$$\mathcal{O} \Longleftrightarrow \bigwedge_{t \in T(\mathcal{O})} \bigvee_{i \in G(\mathcal{O})} \mathbf{a}_i' \mathbf{x}_t \geq b_i, \tag{3}$$

where again $G(\mathcal{O})$ is the set containing the indices of the linear constraints defining the obstacle, and $T(\mathcal{O})$ is the set of time steps at which the obstacle must be avoided. In this notation we use $\wedge$ to denote logical AND and $\vee$ to denote logical OR. Stay-in constraints and obstacles are illustrated in Figures 1 and 2. We state the probabilistic path planning problem as follows:

> *Given a probability distribution for the initial vehicle position, and given a desired goal position, design a finite, optimal sequence of control inputs $\mathbf{u}_0 \ldots \mathbf{u}_{k-1}$ such that the expected final vehicle position corresponds to the goal position, and such that the probability that the vehicle leaves a stay-in region or collides with an obstacle is at most Δ.*

The problem is defined formally here:

Problem 1 (Chance-constrained path planning problem).

[1]Note that nonconvex obstacles can be created by composing several convex obstacles.

$$\min_{\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}} g(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}, \bar{\mathbf{x}}_0, \ldots, \bar{\mathbf{x}}_k) \tag{4}$$

subject to: (5)

$$\bar{\mathbf{x}}_k = \mathbf{x}_{goal} \tag{6}$$

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \omega_t \tag{7}$$

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, P_0) \quad \omega_t \sim \mathcal{N}(0, Q) \tag{8}$$

$$P\left(\left(\bigwedge_{i=1}^{N_{\mathcal{I}}} \mathcal{I}_i \right) \wedge \left(\bigwedge_{j=1}^{N_{\mathcal{O}}} \mathcal{O}_j \right) \right) \geq 1 - \Delta, \tag{9}$$

where $g(\cdot)$ is a piecewise linear cost function such as time or fuel, and $N_{\mathcal{I}}$ and $N_{\mathcal{O}}$ are the number of stay-in regions and obstacles respectively. We use $\bar{\mathbf{v}}$ to denote the expectation of $\mathbf{v}$. We assume throughout this paper that $\Delta \leq 0.5$.

The key difficulty in solving Problem 1 is the non-convex chance constraint (9). There are two difficulties in handling this constraint. First, evaluating the chance constraint requires the computation of the integral of a multi-variable Gaussian distribution over a finite, non-convex region. This cannot be carried out in closed form, and approximate techniques such as sampling are time-consuming and introduce approximation error. Second, even if this integral could be computed efficiently, its value is non-convex in the decision variables due to the disjunctions in $\mathcal{O}_i$. This means that the resulting optimization problem is, in general, intractable. A typical approach to dealing with non-convex feasible spaces is the branch and bound method, which decomposes a non-convex problem into a tree of convex problems. However the branch and bound method cannot be directly applied, since the non-convex chance constraint cannot be decomposed trivially into subproblems.

In order to overcome these two difficulties, we propose a bounding approach to decompose the non-convex chance constraint conservatively into a set of individual chance constraints, each of which is defined on a univariate probability distribution. Integrals over univariate probability distributions can be evaluated accurately and efficiently, and the decomposition of the chance constraint enables the branch and bound algorithm to be applied to find the optimal solution.

IV. Existing Results

In this section we provide some important definitions and review known results to be used later.

A. Disjunctive Convex Programs

Problem 2. *A* Disjunctive Convex Program *is defined as:*

$$\min_X h(X)$$

subject to:

$$f_{eq}(X) = \mathbf{0}$$
$$\bigwedge_{i=1}^{n_{dis}} \bigvee_{j=1}^{n_{cl}} c_{ij}(X) \le 0, \tag{10}$$

where $f_{eq}(X)$ is a linear function of X, the $c_{ij}(X)$ are convex functions of X, and n_{dis} and n_{cl} are the number of disjunctions and clauses within each disjunction, respectively.

Problem 3. *A* Disjunctive Linear Program *is defined as (10) where $f_{eq}(X)$ and $c_{ij}(X)$ are linear functions of X.*

The key difficulty in solving a disjunctive convex program is that the disjunctions in (10) render the feasible region nonconvex. Nonconvex programs are, in general, intractable. In the case of a disjunctive convex program, however, we can decompose the overall optimization problem into a finite number of subproblems that are convex programs. Convex programs can be solved to global optimality with analytic bounds on the number of iterations required for convergence[42], [43]. The number of convex subproblems is exponential in the number of disjunctions n_{dis}. However, for many practical problems, a branch-and-bound approach has been shown to find the globally optimal solution while solving only a small subset of those required in the theoretical worst case[44], [45]. In Appendix A we describe a branch-and-bound approach for solving a general disjunctive convex program.

B. Propagation of Linear Gaussian Statistics

In this paper we assume that the initial state has a Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_0, P_0)$, that the system dynamics are linear, and that there are additive Gaussian white noise processes corresponding to model uncertainty and disturbances. Under these assumptions, the distribution of the future state is also Gaussian, i.e. $p(X_t|\mathbf{u}_{0,\dots,k-1}) \sim \mathcal{N}(\mu_t, \Sigma_t)$. By recursive application of the system equations the distribution of the future state can be calculated exactly as:

$$\mu_t = \sum_{i=0}^{t-1} A^{t-i-1} B \mathbf{u}_i + A^t \hat{\mathbf{x}}_0 \tag{11}$$

$$\Sigma_t = \sum_{i=0}^{t-1} A^i Q (A^T)^i + A^t P_0 (A^T)^t. \tag{12}$$

There are two important properties to note here:

1) The equation for the mean of the state at time t is linear in the control inputs $\mathbf{u}_0, \dots, \mathbf{u}_{t-1}$.

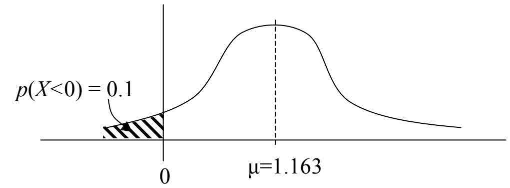

Fig. 3. Univariate Gaussian distribution with mean μ and variance 1. For fixed variance, the chance constraint $p(X < 0) \le 0.1$ is satisfied if and only if $\mu \ge 1.163$

2) The covariance of the state at time t is not a function of the the control inputs $\mathbf{u}_0, \dots, \mathbf{u}_{t-1}$. This means that for a given initial state covariance, and with known noise covariances, the covariance at a future time is known exactly.

These two properties enable the obstacle avoidance problem to be framed as a Disjunctive Convex Program, as will be shown in Section VI.

C. Linear Chance Constraints as Deterministic Linear Constraints

In this section, we show that linear chance constraints on the state of the vehicle at time t, can be expressed *exactly* as deterministic linear constraints on the mean of the vehicle state at time t [10]. In general, a chance constraint on a singlevariate Gaussian random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ with fixed variance but variable mean, can be translated into a deterministic constraint on the mean:

$$p(X < 0) \le \delta \Longleftrightarrow \mu \ge c \tag{13}$$

This is illustrated in Figure 3. The value of the deterministic constraint c is calculated as follows:

$$c = \sqrt{2}\sigma \cdot \mathrm{erf}^{-1}(1 - 2\delta), \tag{14}$$

where erf is defined as:

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt. \tag{15}$$

The inverse of erf can be calculated using a look-up method. Note that only one look-up table is required for any Gaussian distribution. For (14) to be valid, we assume that the probability δ is less than 0.5.

Now consider the case of a multivariate Gaussian random variable X_t, corresponding to the position of the vehicle at time t, which has mean μ_t and covariance Σ_t, and the linear chance constraint $p(\mathbf{a}^T X_t < b) \le \delta$. The event $\mathbf{a}^T X_t < b$ is equivalent to the event $V < 0$, where V is the singlevariate random variable that corresponds to the perpendicular distance between the constraint $\mathbf{a}^T \mathbf{x} = b$ and $\mathbf{x}$, as shown in Fig. 4.

The random variable V is a derived variable of the multivariate random variable X_t. It can be shown that V is a univariate Gaussian random variable, with mean μ_v and variance σ_v, where:

$$\mu_v = \mathbf{a}^T \mu_t - b, \tag{16}$$

and

$$\sigma_v = \sqrt{\mathbf{a}^T \Sigma_t \mathbf{a}}. \tag{17}$$

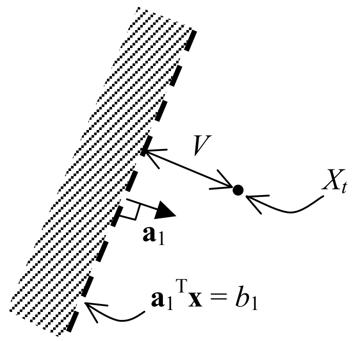

Fig. 4. Linear constraint and vehicle position X_t. V is the distance between the constraint and the vehicle, defined as positive for values of X_t for which the constraint is satisfied, and negative for value of X_t for which the constraint is violated. The vector $\mathbf{a}_1$ is the unit normal in the direction of positive V.

The linear chance constraint $p(\mathbf{a}^T X_t < b) \leq \delta$ is therefore equivalent to a chance constraint $p(V < 0) \leq \delta$ on the univariate Gaussian random variable V. This can be expressed as a deterministic constraint on the mean, of the form $\mu_v \geq c$, where c is given by (14), with $\sigma = \sigma_v$.

Expressing this deterministic constraint in terms of the original variable X_t yields:

$$p(\mathbf{a}^T X_t < b) \leq \delta \Longleftrightarrow \mathbf{a}^T \mu_t - b \geq c, \tag{18}$$

where:

$$c = \sqrt{2\mathbf{a}^T \Sigma_t \mathbf{a}} \cdot \mathrm{erf}^{-1}(1 - 2\delta). \tag{19}$$

This calculation requires knowledge of Σ_t, the covariance of the state at time t. In Section IV-B we showed that Σ_t does not depend on the control inputs, and therefore given an initial state covariance and the noise process covariances, we can *a priori* calculate Σ_t using (12). Furthermore, the right hand side of (18) is linear in the mean μ_t. Hence linear chance constraints on vehicle state can be expressed, *without approximation*, as deterministic linear constraints on the mean of the vehicle state.

V. Disjunctive Convex Bound for Nonconvex Feasible Regions

In this section we describe our main technical result, a novel bound that enables the chance constrained path planning problem (Problem 1) to be approximated as a Disjunctive Convex Program. This extends the bound in [46] from convex polytopic feasible regions to nonconvex polytopic feasible regions.

Lemma 1 (Disjunctive Convex Bound).

$$\begin{aligned} &P\Bigg(\Big(\bigwedge_{i=1}^{N_{\mathcal{I}}} \mathcal{I}_i\Big) \wedge \Big(\bigwedge_{j=1}^{N_{\mathcal{O}}} \mathcal{O}_j\Big)\Bigg) \geq 1 - \Delta \\ \Longleftarrow &\Bigg(\bigwedge_{j=1}^{N_{\mathcal{I}}} \bigwedge_{t \in T(\mathcal{I}_j)} \bigwedge_{i \in G(\mathcal{I}_j)} b_i - \mathbf{a}_i^T \mu_t \geq c_{t,i}\Big(\delta(\mathcal{I}_j, t, i)\Big)\Bigg) \\ &\wedge \\ &\Bigg(\bigwedge_{j=1}^{N_{\mathcal{O}}} \bigwedge_{t \in T(\mathcal{O}_j)} \bigvee_{i \in G(\mathcal{O}_j)} \mathbf{a}_i^T \mu_t - b_i \geq c_{t,i}\Big(\delta(\mathcal{O}_j, t)\Big)\Bigg) \\ &\wedge \\ &\sum_{j=1}^{N_{\mathcal{I}}} \sum_{t \in T(\mathcal{I}_j)} \sum_{i \in G(\mathcal{I}_j)} \delta(\mathcal{I}_j, t, i) + \sum_{j=1}^{N_{\mathcal{O}}} \sum_{t \in T(\mathcal{O}_j)} \delta(\mathcal{O}_j, t) \leq \Delta, \end{aligned} \tag{20}$$

where:

$$c_{t,i}(\xi) = \sqrt{2\mathbf{a}_i^T \Sigma_t \mathbf{a}_i} \cdot \mathit{erf}^{-1}\Big(1 - 2\xi\Big). \tag{21}$$

Proof: Using DeMoivre's theorem we have:

$$P\Bigg(\Big(\bigwedge_{i=1}^{N_{\mathcal{I}}} \mathcal{I}_i\Big) \wedge \Big(\bigwedge_{j=1}^{N_{\mathcal{O}}} \mathcal{O}_j\Big)\Bigg) = 1 - P\Bigg(\Big(\bigwedge_{i=1}^{N_{\mathcal{I}}} \tilde{\mathcal{I}}_i\Big) \vee \Big(\bigwedge_{j=1}^{N_{\mathcal{O}}} \tilde{\mathcal{O}}_j\Big)\Bigg), \tag{22}$$

where $\tilde{A}$ denotes the logical complement of event A. From the definitions of $\mathcal{I}$ and $\mathcal{O}$ we can write:

$$\begin{aligned} \tilde{\mathcal{I}}_i &\Longleftrightarrow \bigvee_{t \in T(\mathcal{I}_j)} \bigvee_{i \in G(\mathcal{I}_j)} \mathbf{a}_i' \mathbf{x}_t > b_i \\ \tilde{\mathcal{O}}_i &\Longleftrightarrow \bigvee_{t \in T(\mathcal{O}_j)} \bigwedge_{i \in G(\mathcal{O}_j)} \mathbf{a}_i' \mathbf{x}_t \leq b_i \end{aligned} \tag{23}$$

Boole's bound shows that for any two events A and B:

$$P(A \vee B) \leq P(A) + P(B), \tag{24}$$

and hence for any number of events A_i we have:

$$P\Big(\bigvee_i A_i\Big) \leq \sum_i P(A_i). \tag{25}$$

In addition, for any two events A and B it is well known that:

$$P(A \wedge B) \leq P(A) \quad P(A \wedge B) \leq P(B), \tag{26}$$

and hence for any number of events A_i we have:

$$P\Big(\bigwedge_i A_i\Big) \leq P(A_j) \quad \forall j. \tag{27}$$

The bounds (25) and (27) form the core of our disjunctive convex bound. Using these two results, we can show that:

$$P\Bigg(\Big(\bigwedge_{i=1}^{N_{\mathcal{I}}} \mathcal{I}_i\Big) \wedge \Big(\bigwedge_{j=1}^{N_{\mathcal{O}}} \mathcal{O}_j\Big)\Bigg) \geq 1 - \sum_{i=1}^{N_{\mathcal{I}}} P(\tilde{\mathcal{I}}_i) - \sum_{j=1}^{N_{\mathcal{O}}} P(\tilde{\mathcal{O}}_j). \tag{28}$$

Applying Boole's bound again, and the result in (23) we have:

$$P\left(\left(\bigwedge_{i=1}^{N_{\mathcal{I}}}\mathcal{I}_i\right)\wedge\left(\bigwedge_{j=1}^{N_{\mathcal{O}}}\mathcal{O}_j\right)\right)\geq$$
$$1-\sum_{j=1}^{N_{\mathcal{I}}}\sum_{t\in T(\mathcal{I}_j)}\sum_{i\in G(\mathcal{I}_j)}P(\mathbf{a}_i'\mathbf{x}_t>b_i)$$
$$-\sum_{j=1}^{N_{\mathcal{O}}}\sum_{t\in T(\mathcal{O}_j)}\Big(P(\mathbf{a}_l'\mathbf{x}_t\leq b_l)\ \forall l\in G(\mathcal{O}_j)\Big). \tag{29}$$

From (18) we know that:

$$\begin{aligned}b_i-\mathbf{a}_i^T\mu_t\geq c_{t,i}(\xi)&\Longrightarrow P(\mathbf{a}_i'\mathbf{x}_t>b_i)\leq\xi\\ \mathbf{a}_i^T\mu_t-b_i\geq c_{t,i}(\xi)&\Longrightarrow P(\mathbf{a}_i'\mathbf{x}_t<b_i)\leq\xi.\end{aligned} \tag{30}$$

Hence:

$$\left(\bigwedge_{j=1}^{N_{\mathcal{I}}}\bigwedge_{t\in T(\mathcal{I}_j)}\bigwedge_{i\in G(\mathcal{I}_j)}b_i-\mathbf{a}_i^T\mu_t\geq c_{t,i}\Big(\delta(\mathcal{I}_j,t,i)\Big)\right)$$
$$\wedge$$
$$\left(\bigwedge_{j=1}^{N_{\mathcal{O}}}\bigwedge_{t\in T(\mathcal{O}_j)}\bigvee_{i\in G(\mathcal{O}_j)}\mathbf{a}_i^T\mu_t-b_i\geq c_{t,i}\Big(\delta(\mathcal{O}_j,t)\Big)\right)$$
$$\wedge$$
$$\sum_{j=1}^{N_{\mathcal{I}}}\sum_{t\in T(\mathcal{I}_j)}\sum_{i\in G(\mathcal{I}_j)}\delta(\mathcal{I}_j,t,i)+\sum_{j=1}^{N_{\mathcal{O}}}\sum_{t\in T(\mathcal{I}_j)}\delta(\mathcal{O}_j,t)\leq\Delta$$
$$\Longrightarrow\quad\exists l(j,t)\text{ such that:}$$
$$\sum_{j=1}^{N_{\mathcal{I}}}\sum_{t\in T(\mathcal{I}_j)}\sum_{i\in G(\mathcal{I}_j)}P(\mathbf{a}_i'\mathbf{x}_t>b_i)$$
$$+\sum_{j=1}^{N_{\mathcal{O}}}\sum_{t\in T(\mathcal{O}_j)}P(\mathbf{a}_{l(j,t)}'\mathbf{x}_t\leq b_{l(j,t)})$$
$$\leq\sum_{j=1}^{N_{\mathcal{I}}}\sum_{t\in T(\mathcal{I}_j)}\sum_{i\in G(\mathcal{I}_j)}\delta(\mathcal{I}_j,t,i)+\sum_{j=1}^{N_{\mathcal{O}}}\sum_{t\in T(\mathcal{O}_j)}\delta(\mathcal{O}_j,t)\leq\Delta$$
$$\Longrightarrow P\left(\left(\bigwedge_{i=1}^{N_{\mathcal{I}}}\mathcal{I}_i\right)\wedge\left(\bigwedge_{j=1}^{N_{\mathcal{O}}}\mathcal{O}_j\right)\right)\geq 1-\Delta, \tag{31}$$

which completes the proof. ∎

In (20) the parameters $\delta(\mathcal{I}_j,t,i)$ and $\delta(\mathcal{O}_j,t)$ are referred to as the *risk* that is allocated to each of the univariate chance constraints. By ensuring that these risks sum to at most Δ, we ensure that the overall probability of failure is at most Δ, as required. We refer to the optimization of the individual risks $\delta(\cdot)$ as *risk allocation*, whereas the process of choosing which of the disjunctions in (20) to satisfy is called *risk selection*.

VI. Path Planning using Disjunctive Convex Programming

In this section we approximate the chance constrained path planning problem (Problem 1) as a Disjunctive Convex Program using the bound introduced in Section V.

Problem 4 (Path Planning as Disjunctive Convex Program).

$$\min_{\mathbf{u}_0,\ldots,\mathbf{u}_{k-1},\delta(\cdot)} g(\mathbf{u}_0,\ldots,\mathbf{u}_{k-1},\bar{\mathbf{x}}_0,\ldots,\bar{\mathbf{x}}_k) \tag{32}$$

subject to (11), (12), (21), and: (33)

$$\bar{\mathbf{x}}_k=\mathbf{x}_{goal} \tag{34}$$
$$\bigwedge_{j=1}^{N_{\mathcal{I}}}\bigwedge_{t\in T(\mathcal{I}_j)}\bigwedge_{i\in G(\mathcal{I}_j)}b_i-\mathbf{a}_i^T\mu_t\geq c_{t,i}\Big(\delta(\mathcal{I}_j,t,i)\Big) \tag{35}$$
$$\bigwedge_{j=1}^{N_{\mathcal{O}}}\bigwedge_{t\in T(\mathcal{O}_j)}\bigvee_{i\in G(\mathcal{O}_j)}\mathbf{a}_i^T\mu_t-b_i\geq c_{t,i}\Big(\delta(\mathcal{O}_j,t)\Big) \tag{36}$$

$$\sum_{j=1}^{N_{\mathcal{I}}}\sum_{t\in T(\mathcal{I}_j)}\sum_{i\in G(\mathcal{I}_j)}\delta(\mathcal{I}_j,t,i)+\sum_{j=1}^{N_{\mathcal{O}}}\sum_{t\in T(\mathcal{O}_j)}\delta(\mathcal{O}_j,t)\leq\Delta \tag{37}$$

$$\delta(\mathcal{I}_j,t,i)\geq 0\qquad\delta(\mathcal{O}_j,t)\geq 0\quad\forall j,t,i. \tag{38}$$

Lemma 2. *Any feasible solution to Problem 4 is a feasible solution to Problem 1.*

Proof: Lemma 1 shows that the constraints (35) through (38) imply the full chance constraint (9). All other constraints are identical between Problem 1 and Problem 4, which completes the proof. ∎

Lemma 3. *For $\Delta\leq 0.5$, Problem 4 is a Disjunctive Convex Program.*

Proof: The fact that $\Delta\leq 0.5$ implies that $\delta(\mathcal{I}_j,t,i)\leq 0.5$ and $\delta(\mathcal{O}_j,t)\leq 0.5$. The function $c_{t,i}(\xi)$ is convex in ξ for $\xi\leq 0.5$. This implies that each of the scalar inequalities:

$$b_i-\mathbf{a}_i^T\mu_t\geq c_{t,i}\Big(\delta(\mathcal{I}_j,t,i)\Big), \tag{39}$$

is convex in the decision variables $\delta(\mathcal{I}_j,t,i)$ and μ_t. Hence the conjunction of inequalities (35) is convex in the decision variables. In addition the scalar inequalities:

$$\mathbf{a}_i^T\mu_t-b_i\geq c_{t,i}\Big(\delta(\mathcal{O}_j,t)\Big), \tag{40}$$

are convex in the decision variables $\delta(\mathcal{O}_j,t)$ and μ_t. Hence (36) is a disjunction of convex inequalities. Since the equality constraints and the inequality (37) are linear, Problem 4 is a Disjunctive Convex Program. ∎

Since Problem 4 is a Disjunctive Convex Program, we can use the existing approaches reviewed in Appendix A to solve it to global optimality in finite time, and from Lemma 2 we know that the resulting sequence of control inputs is guaranteed to be feasible for the original chance constrained path planning problem. Solution of the full Disjunctive Convex Program, however, is slow. In the following sections we propose an approach to solve Problem 4 approximately, in order to reduce the computation time dramatically.

VII. Fixed Risk Tightening

In this section we provide a tightening of Problem 4, where each of the risks allocated to each chance constraint has a fixed value. This *fixed risk tightening* (FRT) is used to give an upper bound on the optimal cost of Problem 4. The tightening is a Disjunctive Linear Program, and hence can be solved efficiently using highly optimized commercial solvers[6].

Problem 5 (Fixed Risk Tightening).

$$\min_{\mathbf{u}_0,\ldots,\mathbf{u}_{k-1}} g(\mathbf{u}_0,\ldots,\mathbf{u}_{k-1},\bar{\mathbf{x}}_0,\ldots,\bar{\mathbf{x}}_k) \tag{41}$$

$$\textit{subject to (11), (12), (21), and:} \tag{42}$$

$$\bar{\mathbf{x}}_k = \mathbf{x}_{goal} \tag{43}$$

$$\left(\bigwedge_{j=1}^{N_\mathcal{I}} \bigwedge_{t\in T(\mathcal{I}_j)} \bigwedge_{i\in G(\mathcal{I}_j)} b_i - \mathbf{a}_i^T\mu_t \geq c_{t,i}(\delta)\right) \tag{44}$$

$$\left(\bigwedge_{j=1}^{N_\mathcal{O}} \bigwedge_{t\in T(\mathcal{O}_j)} \bigvee_{i\in G(\mathcal{O}_j)} \mathbf{a}_i^T\mu_t - b_i \geq c_{t,i}(\delta)\right) \tag{45}$$

$$\delta = \frac{\Delta}{\sum\limits_{j=1}^{N_\mathcal{I}} \sum\limits_{t\in T(\mathcal{I}_j)} |G(\mathcal{I}_j)| + \sum\limits_{j=1}^{N_\mathcal{O}} |T(\mathcal{O}_j)|} \tag{46}$$

$$c_{t,i}(\delta) = \sqrt{2\mathbf{a}_i^T\Sigma_t\mathbf{a}_i}\cdot \mathit{erf}^{-1}(1-2\delta). \tag{47}$$

We use $|\cdot|$ to denote the number of elements in a set. The main difference between Problem 5 and Problem 4 is that in Problem 5 the risks associated with violation of each constraint are *a priori* set to a fixed and equal value δ, given by (46), whereas in Problem 4 the risks are optimization variables.

Lemma 4. *Any feasible solution to Problem 5 is a feasible solution to Problem 1.*

Proof: Setting $\delta(\mathcal{I}_j,t,i) = \delta(\mathcal{O}_j,t,i) = \delta$ in (20) and noting that:

$$\sum_{j=1}^{N_\mathcal{I}} \sum_{t\in T(\mathcal{I}_j)} \sum_{i\in G(\mathcal{I}_j)} \delta + \sum_{j=1}^{N_\mathcal{O}} \sum_{t\in T(\mathcal{O}_j)} \delta = \Delta, \tag{48}$$

we see that Lemma 1 shows that the constraints (44) through (47) imply the full chance constraint (9). All other constraints are identical between Problem 1 and Problem 5, which completes the proof. ■

Lemma 5. *A solution to Problem 5 is a feasible solution to Problem 4, and the optimal solution to Problem 5 has cost greater than, or equal to, the cost of the optimal solution to Problem 4.*

Proof: Comparison of the constraints in Problem 4 and Problem 5 shows that the feasible set of Problem 5 is contained in the feasible set of Problem 4, from which the results follow. ■

Problem 5 is a Disjunctive Linear Program since we have assumed a linear cost, the equality constraints are linear in the decision variables $\mathbf{u}_0,\ldots,\mathbf{u}_{k+1}$, and the disjunctive constraints (44) through (47) are linear in the decision variables.

VIII. Fixed Risk Relaxation

In this section we provide a relaxation of the full Disjunctive Convex Program given in Problem 4, which we call the *fixed risk relaxation* (FRR). As in Section VII, each of the each of the risks allocated to each chance constraint has a fixed value, however the fixed value is now chosen so that the new approximation is a relaxation of Problem 4. We use this relaxation to provide a lower bound on the optimal cost of Problem 4. Since the relaxation given in this section is a Disjunctive Linear Program, it can be solved efficiently using highly optimized commercial solvers.

Problem 6 (Fixed Risk Relaxation).

$$\min_{\mathbf{u}_0,\ldots,\mathbf{u}_{k-1}} g(\mathbf{u}_0,\ldots,\mathbf{u}_{k-1},\bar{\mathbf{x}}_0,\ldots,\bar{\mathbf{x}}_k) \tag{49}$$

$$\textit{subject to (11), (12), (21), and:} \tag{50}$$

$$\bar{\mathbf{x}}_k = \mathbf{x}_{goal} \tag{51}$$

$$\left(\bigwedge_{j=1}^{N_\mathcal{I}} \bigwedge_{t\in T(\mathcal{I}_j)} \bigwedge_{i\in G(\mathcal{I}_j)} b_i - \mathbf{a}_i^T\mu_t \geq c_{t,i}(\Delta)\right) \tag{52}$$

$$\left(\bigwedge_{j=1}^{N_\mathcal{O}} \bigwedge_{t\in T(\mathcal{O}_j)} \bigvee_{i\in G(\mathcal{O}_j)} \mathbf{a}_i^T\mu_t - b_i \geq c_{t,i}(\Delta)\right) \tag{53}$$

$$c_{t,i}(\Delta) = \sqrt{2\mathbf{a}_i^T\Sigma_t\mathbf{a}_i}\cdot \mathit{erf}^{-1}(1-2\Delta). \tag{54}$$

The key difference between Problem 6 and Problem 5 is that in Problem 6 the risks associated with violation of each constraint are *a priori* set to a fixed and equal value Δ, whereas in Problem 5 they are set to a smaller value given by (46).

Lemma 6. *The optimal cost obtained by solving Problem 6 is a lower bound on the optimal cost obtained by solving Problem 4.*

Proof: It follows from (37) and (38) that $\delta(\mathcal{I}_j,t,i)\leq\Delta$ and $\delta(\mathcal{O}_j,t)\leq\Delta\ \forall j,t,i$. Since $c_{t,i}(\xi)$ is a monotonically decreasing function of ξ, all scalar chance constraints in (52) and (53) of Problem 6 are looser than the scalar chance constraints in (35) and (36) of Problem 4. Therefore, the cost of the optimal solution of Problem 6 is less than or equal to the optimal cost of Problem 4. ■

The Fixed Risk Relaxation (Problem 6) is a Disjunctive Linear Program.

IX. Solving the Disjunctive Convex Program

The Disjunctive Convex Program for Path Planning given in Problem 4 can be solved to global optimality using existing branch and bound techniques. In Appendix A we give an overview of these techniques. Applying this approach to Problem 4, however, is time-consuming, since it requires the solution of a (potentially) large number of nonlinear convex programs. Hence its applicability to onboard trajectory generation is limited. Instead, we propose in this section a customized solution method that, using the bounds in Sections VII and VIII, dramatically improves computation speed while providing solutions that are suboptimal by a known amount. In Section XI we show empirically that, for a UAV path planning example, the level of suboptimality is very small in almost all cases.

A. Customized Approach

The main idea behind the customized approach is to solve first the fixed risk relaxation and the fixed risk tightening to provide, respectively, lower and upper bounds on the optimal cost of Problem 4, and then use the returned solutions to identify convex regions in which to look for improved solutions. The approach requires the solution of, at most, two Disjunctive Linear Programs and two Convex Programs, and hence can be solved quickly to global optimality.

First, we introduce another tightening of Problem 4.

Problem 7 (Risk Allocation in Convex Region).

$$\min_{\mathbf{u}_0,\ldots,\mathbf{u}_{k-1},\delta(\cdot)} g(\mathbf{u}_0,\ldots,\mathbf{u}_{k-1},\bar{\mathbf{x}}_0,\ldots,\bar{\mathbf{x}}_k) \tag{55}$$

subject to: (56)

$$\bar{\mathbf{x}}_k = \mathbf{x}_{goal} \tag{57}$$

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \omega_t \tag{58}$$

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, P_0) \quad \omega_t \sim \mathcal{N}(0, Q) \tag{59}$$

$$\left(\bigwedge_{j=1}^{N_{\mathcal{I}}} \bigwedge_{t\in T(\mathcal{I}_j)} \bigwedge_{i\in G(\mathcal{I}_j)} b_i - \mathbf{a}_i^T\mu_t \geq c_{t,i}\Big(\delta(\mathcal{I}_j,t,i)\Big)\right) \tag{60}$$

$$\left(\bigwedge_{j=1}^{N_{\mathcal{O}}} \bigwedge_{t\in T(\mathcal{O}_j)} \mathbf{a}_{H(t,\mathcal{O}_j)}^T\mu_t - b_{H(\mathcal{O}_j)} \geq c_{t,H(t,\mathcal{O}_j)}\Big(\delta(\mathcal{O}_j,t)\Big)\right) \tag{61}$$

$$\sum_{j=1}^{N_{\mathcal{I}}} \sum_{t\in T(\mathcal{I}_j)} \sum_{i\in G(\mathcal{I}_j)} \delta(\mathcal{I}_j,t,i) + \sum_{j=1}^{N_{\mathcal{O}}} \sum_{t\in T(\mathcal{O}_j)} \delta(\mathcal{O}_j,t) \leq \Delta \tag{62}$$

$$\delta(\mathcal{I}_j,t,i) \geq 0 \quad \delta(\mathcal{O}_j,t) \geq 0 \quad \forall j,t,i. \tag{63}$$

Here $H(t,\mathcal{O})$ is a mapping from an obstacle-timestep pair $\{\mathcal{O},t\}$ to the index of a single constraint within obstacle $\mathcal{O}$. The main difference between Problem 4 and Problem 7 is that in Problem 7, the disjunction of constraints has been replaced with a single constraint to be satisfied for each obstacle, for each time step. As a consequence, Problem 7 is a Convex Program, rather than a Disjunctive Convex Program, and Problem 7 is a tightening of Problem 4.

Lemma 7. *Any feasible solution to Problem 7 is a feasible solution to Problem 4, and the optimal cost of Problem 7 is an upper bound on the optimal cost to Problem 4.*

Proof: By comparison of constraints, we see that Problem 7 is a tightening of Problem 4, from which the proof follows. ∎

The customized solution approach is as follows:

Algorithm 1 (Customized solution approach).

1) *Set $J_{UB} = +\infty$, $J_{LB} = -\infty$ and $\mathtt{sol}^* = \emptyset$.*
2) *Solve fixed risk relaxation (Problem 6). If infeasible, set $J_{LB} = +\infty$ and stop, else assign optimal cost to J_{LB}.*
3) *Assign to $H(t,\mathcal{O}_j)$ the set of constraints in (53) that were, for each timestep t, for each obstacle $\mathcal{O}_j$, satisfied with the greatest margin in Problem 6.*
4) *Solve risk allocation in convex region (Problem 7). If feasible, assign optimal cost to J_{UB}, assign optimal solution to $\mathtt{sol}^*$ and stop.*
5) *Solve fixed risk tightening (Problem 5). If feasible, assign optimal cost to J_{UB} and assign optimal solution to $\mathtt{sol}^*$, otherwise set $J_{UB} = +\infty$ and stop.*
6) *Assign to $H(t,\mathcal{O}_j)$ the set of constraints in (45) that were, for each timestep t, for each obstacle $\mathcal{O}_j$, satisfied with the greatest margin in Problem 5.*
7) *Solve risk allocation in convex region (Problem 7). Assign optimal cost to J_{UB} and optimal solution to $\mathtt{sol}^*$.*

Theorem 1. *Upon termination of the customized approach, any solution $\mathtt{sol}^*$ returned is a feasible solution to Problem 4 and $J_{LB} \leq J^* \leq J_{UB}$, where J^* is the optimal cost to Problem 4. In addition, $\mathtt{sol}^*$ is a feasible solution to Problem 1, and J_{UB} is an upper bound on the optimal cost for Problem 1.*

Proof: From Lemmas 5 and 7 we know that the solutions assigned to $\mathtt{sol}^*$ in Steps 4, 5 and 7 are feasible solutions to Problem 4, and hence all of the values assigned to J_{UB} are upper bounds on J^*. From Lemma 6 we know that the value assigned to J_{LB} in Step 2 is a lower bound on J^*. From Lemma 2, the rest of the proof follows. ∎

The customized solution approach therefore provides bounds on the suboptimality of the solution that it returns.

In Steps 4 and 7 we choose to solve the risk allocation in the convex region defined by the set of constraints that were satisfied by the greatest margin in Steps 2 and 5, respectively. This is a heuristic that we claim leads to good solutions in most cases. When allocating risk from the fixed risk tightening, the goal is to start from a feasible upper bound on the optimal cost and push the state mean closer to the obstacles to reduce cost. Hence choosing constraints that the mean is furthest away from is reasonable heuristic. Similarly, when allocating risk from the fixed risk relaxation, the goal is to find a feasible solution starting from an infeasible lower bound on the cost. Hence choosing constraints such that there is the greatest possible margin to each constraint is an intuitive approach.

B. Discussion of customized approach

Unlike the full branch and bound algorithm given in Appendix A, it is possible for the customized approach to return no solution, when one does exist, along with the trivial bounds $-\infty \leq J^* \leq \infty$. Note, however, that this only occurs if the fixed risk relaxation (Step 2) returns a feasible solution, but none of the subsequent steps returns a feasible solution. Intuitively, this occurs only when an optimal solution exists, but in a different convex region from both the fixed risk relaxation and the fixed risk tightening. In Section XI we show that, for a UAV path planning problem, this situation occurs very infrequently in practice By the same token, it is possible for the algorithm to return very loose (but nontrivial) bounds on J^*. Intuitively, this occurs when the fixed risk relaxation step finds a solution in a different convex region than the solution found in the fixed risk tightening step, but no feasible solution is found in Step 4. This occurs when there are very different corridors through the obstacle field, at least two of which have probabilities of failure close to the requirement Δ. Again, in Section XI we show that, for a UAV path planning problem, this situation occurs very infrequently in practice,

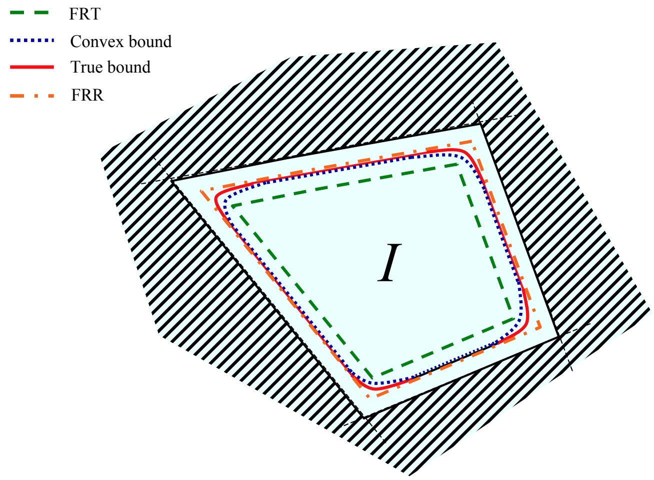

Fig. 5. Stay-in region from Figure 1 and illustration of bounds. The lines show the feasible regions for a given value of Δ for the various bounds introduced in this paper.

and that J_{LB} and J_{UB} are very close in almost all cases. Hence in almost all cases the customized solution approach either correctly identifies that Problem 4 is infeasible, or returns a solution that is very close to the optimal solution to Problem 4.

X. Illustration of Bounds

The various bounds introduced in this paper are illustrated for a single timestep for the case of a single stay-in region in Figure 5 and for the case of a single obstacle in Figure 6. In Figure 5 it can be seen that the FRT used in Problem 5 has the smallest feasible region, since it is the most conservative. This feasible region is generated by backing off from each of the linear constraints so that each is violated with probability at most $\Delta/|\mathcal{G}(\mathcal{I})|$. Note that in the figure, the backoff distance is equal for all constraints indicating that $p(\mathbf{x}_t)$ is symmetric, but this is not necessarily true in general. The convex bound used in Problem 4 is significantly less conservative. This bound uses risk allocation to assign different risks to each of the linear constraints. Away from the corners of the stay-in region, virtually all of the available risk Δ is allocated to a single constraint. Since in a one-dimensional case, the conversion from the chance constraint to a deterministic constraint on the mean is exact, away from the corners the feasible region with the convex bound is very close to the feasible region of the true problem (Problem 1) shown in red. At the corners, however, the convex bound still has noticeable conservatism. This is because, at the corners, a similar amount of risk is allocated to more than one linear constraint, and in doing so we incur the conservatism of Boole's bound. Finally, it can be seen that the Fixed Risk Relaxation (Problem 6) has a larger feasible region than the convex problem since it is a relaxation of Problem 4, and that the feasible region is a polytope since the FRR is a linear program. In Figure 6 it can be seen that the FRT is again conservative, and gives a polytopic feasible region as expected. For a single obstacle and time step, the convex bound, FRT and FRR are identical. The FRT feasible region is generated by backing off from each of the edges of the obstacle such that the probability of violation of a single linear

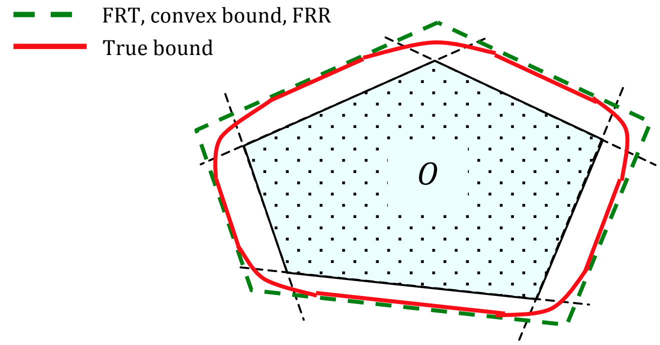

Fig. 6. Obstacle from Figure 2 and illustration of bounds. The lines show the feasible regions for a given value of Δ for the various bounds introduced in this paper. Note that since we have a single obstacle and time step, the convex bound, FRT and FRR are identical.

constraint is at most Δ. Away from the corners of the obstacle, the probability of collision with the obstacle is very close to the probability of violation of a single linear constraint. Hence the true feasible region is close to the FRT feasible region away from the corners, as can be seen in Figure 6. Again the Fixed Risk Relaxation (Problem 6) has a larger feasible region than the true problem and the feasible region is a polytope since the FRR is a linear program.

XI. Simulation Results

In this section, we demonstrate, in simulation, the new method for chance-constrained path planning, using a UAV path planning example. In this example a UAV is operating in a two-dimensional obstacle field, subject to uncertain localization and wind disturbances. The UAV must plan a path from its initial state, to a goal location, subject to a chance constraint that ensures the probability of collision with any obstacle is at most Δ. This problem is an example of the chance-constrained path planning problem (Problem 1). We solve this using the customized solution approach (Algorithm 1), which involves the solution of the FRT (Problem 5), the FRR (Problem 6), and risk allocation in convex regions (Problem 7). We use the YALMIP interface for Matlab[47], with CPLEX[6] used to solve the MILPs generated by the FRR and FRT steps, and with SNOPT[48] used to solve the nonlinear convex programs generated by the risk allocation steps.

The UAV is modeled as a double integrator with an inner-loop velocity controller and maximum velocity constraints, as proposed by [5]. The state and control inputs to this system are defined by:

$$\mathbf{x}_t \triangleq \begin{bmatrix} p_t^x \\ v_t^x \\ p_t^y \\ v_t^y \end{bmatrix} \quad \mathbf{u} \triangleq \begin{bmatrix} v_{t,des}^x \\ v_{t,des}^y \end{bmatrix}, \tag{64}$$

where p_t^x and p_t^y are the positions of the UAV along the x and y axes at time t, v_t^x and v_t^y are the UAV velocities at time t, and the desired velocities to be tracked by the velocity controller are denoted $v_{t,des}^x$ and $v_{t,des}^y$. We implement a maximum velocity constraint of 3m/s as a constraint on the 2-norm of the mean velocity, approximated as a 32-sided polygon. This is illustrated in Figure 7. We discretize the closed-loop UAV

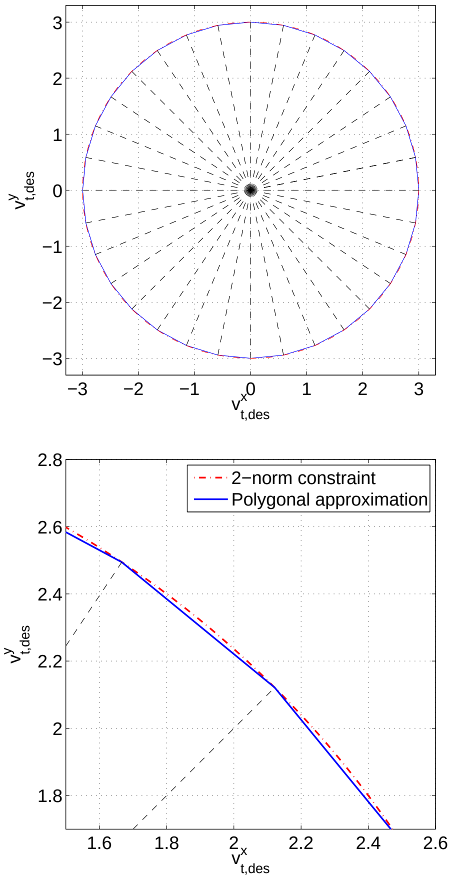

Fig. 7. Approximation of 2-norm constraint as 32-sided polygon. This approximation is used both in bounding the velocity of the UAV to 3m/s (as shown here) and in computing the cost of the planned path, which is proportional to the 2-norm of velocity.

dynamics with a time step of $\Delta t = 1$s, giving the dynamics:

$$A = \begin{bmatrix} 1 & 0.7869 & 0 & 0 \\ 0 & 0.6065 & 0 & 0 \\ 0 & 0 & 1 & 0.7869 \\ 0 & 0 & 0 & 0.6065 \end{bmatrix} \quad B = \begin{bmatrix} 0.2131 & 0 \\ 0.3935 & 0 \\ 0 & 0.2131 \\ 0 & 0.3935 \end{bmatrix}. \tag{65}$$

The cost function $g(\cdot)$ is defined to be proportional to the magnitudes of the commanded velocities, such that:

$$g(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}, \bar{\mathbf{x}}_0, \ldots, \bar{\mathbf{x}}_k) \triangleq \sum_{t=0}^{k-1} \left\| \begin{bmatrix} v^x_{t,des} \\ v^y_{t,des} \end{bmatrix} \right\|_2, \tag{66}$$

where the 2-norm is approximated using a 32-sided polygon as in Figure 7. The initial state of the UAV is modeled as a Gaussian random variable as in Problem 1, where:

$$\hat{\mathbf{x}}_0 \triangleq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad P_0 \triangleq \begin{bmatrix} 0.05^2 & 0 & 0 & 0 \\ 0 & 0.0005^2 & 0 & 0 \\ 0 & 0 & 0.05^2 & 0 \\ 0 & 0 & 0 & 0.0005^2 \end{bmatrix}. \tag{67}$$

The disturbance process ω_t is modeled as a zero-mean Gaussian with covariance:

$$Q \triangleq \begin{bmatrix} 0.3555 & 0 & 0 & 0 \\ 0 & 0.6320 & 0 & 0 \\ 0 & 0 & 0.3555 & 0 \\ 0 & 0 & 0 & 0.6320 \end{bmatrix}. \tag{68}$$

For all maps in this section we use $k = 20$ time steps and the goal is given by:

$$\mathbf{x}_{goal} \triangleq \begin{bmatrix} 0 \\ 10 \end{bmatrix}. \tag{69}$$

In Section XI-A we give results for a single obstacle field, designed to be challenging for a chance-constrained path planning algorithm. In Section XI-B we give statistical results using randomized obstacles fields.

A. Hard map results

In this section we consider the map shown in Figure 8. This map is chosen in order to be challenging for the chance-constrained path planning algorithm, because for certain values of Δ there is a path through the narrow corridor at $(-1, 5)$ that barely satisfies the chance constraint[2]. Figure 8 shows an example of the path planned by the customized solution algorithm for $\Delta = 0.001$. For this particular path, the cost is 10.37, the estimated probability of collision is 4.6×10^{-4}, which satisfies the chance constraint, and the solution time is 16.4s. The lower bound on the cost is 10.27, hence the suboptimality introduced by using the customized approach, rather than full Disjunctive Convex Programming, is at most 0.97%. Figure 9 shows the planned path for a range of different values of Δ. As the user-specified allowable probability of collision decreases, the planned path becomes more conservative, and eventually changes from passing through the narrow corridor to taking a significantly longer, but safer, path around the outside. Figure 10 shows that, as Δ decreases, the cost increases monotonically. This illustrates an important property of chance-constrained planning, where the operator can trade optimality against conservatism.

Figure 11 illustrates how the customized solution for chance constrained path planning described in Section IX operates. In this case we set $\Delta = 0.001$. Figure 11 shows that the fixed risk relaxation (Step 2) finds a path through the corridor, however since this is a relaxed solution, we do not know that it satisfies the chance constraint. In Step 4 the method then uses risk allocation to find a solution that does satisfy the chance constraint, in the vicinity of the fisked risk relaxation. For illustration we also show the solutions that would be given by the fixed risk tightening (Step 5) and risk allocation in the vicinity of the fisked risk tightening (Step 7). While both of these solutions are feasible for the chance-constrained path planning problem, the conservatism of the bounds used means they fail to find the path through the corridor, even though one exists. This example is chosen to illustrate the value of using the fixed risk relaxation to initialize the search for a feasible solution. Figure 12 compares the true probability of collision,

[2]Note that a map for which no feasible path exists is easier for the path planning algorithm, since that infeasibility can typically be detected quickly.

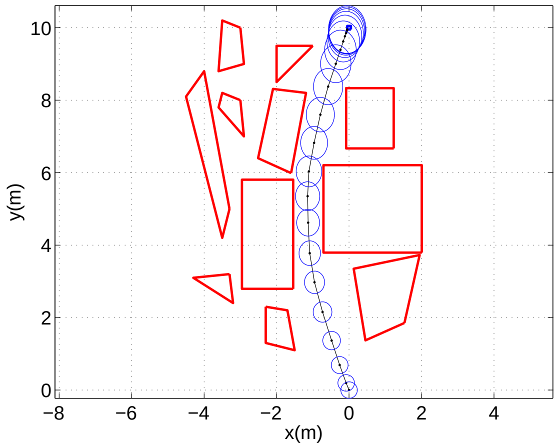

Fig. 8. Example solution with hard map and $\Delta = 0.001$. The goal is shown as the small blue square at $(0, 10)$. The dots show the mean of the planned UAV position at each time step, and the ellipses give the 99% certainty region for the position.

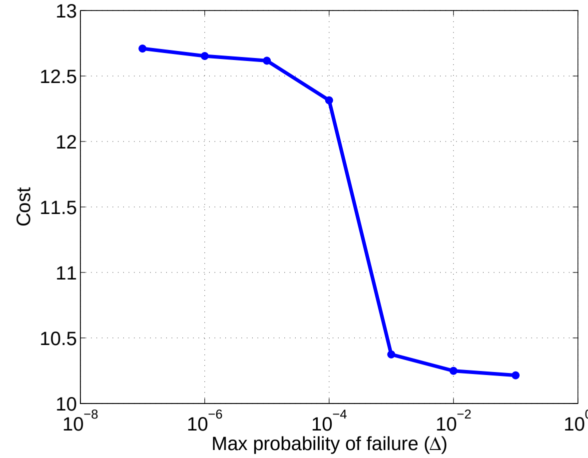

Fig. 10. Cost of returned solution against Δ. As the maximum allowable probability of failure specified by the operator increases, the cost decreases monotonically. The sharp drop corresponds to finding the path through the narrow corridor.

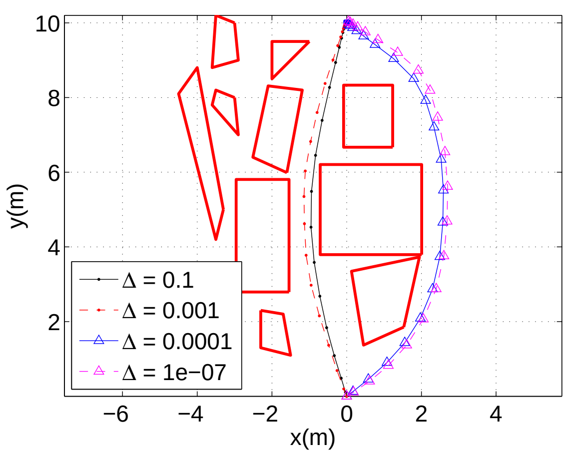

Fig. 9. Solutions with hard map for range of different Δ values. The dots show the mean of the planned UAV position at each time step.

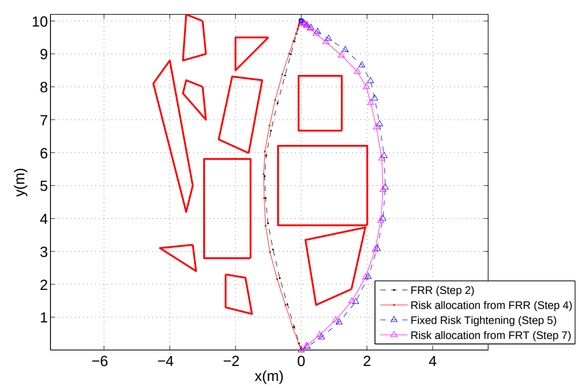

Fig. 11. Illustration of custom solution method for $\Delta = 0.001$. Fixed risk relaxation finds a path through the corridor, which may not satisfy the chance constraint. Risk allocation is able to find a solution that does satisfy the chance constraint, in the vicinity of the fisked risk relaxation. The solutions given by the fixed risk tightening and risk allocation in the vicinity of the fisked risk tightening fail to find the path through the corridor, even though one exists.

estimated using 10^6 Monte Carlo simulations, with the maximum allowable probability of failure Δ. This shows that the bounding approach introduced in this paper is conservative by a factor of approximately 2, and does not depend on Δ. This conservatism is many orders of magnitude better than the values reported in [49], [18], [50], [33] on the bounding approaches of [30], [34], [35], [36], [29]. Furthermore those approaches apply only to convex feasible regions, unlike the new disjunctive convex bound.

To compare the computational efficiency of the customized solution approach against full Disjunctive Convex Programming, we posed the same chance-constrained path planning problem as a full Disjunctive Convex Program (Problem 4). The YALMIP branch-and-bound solver was used with SNOPT as the solver for the convex subproblems. First, we solved the problem removing all obstacles except the one centered at $(0.5, 5)$. In this case the optimal solution was found in 112.04s. Then we attempted to solve the problem with all obstacles, however in this case the full Disjunctive Convex Programming approach was unable to find the optimal solution in 2×10^3s of computation time. This demonstrates that the customized solution approach is at least two orders of magnitude more computationally efficient than solving the full disjunctive convex program, and we show in Section XI-B that the suboptimality introduced is very low in most cases.

B. *Randomized map results*

In this section we perform a statistical study over 500 randomly generated maps. The maps each contain 10 square obstacles, with the center of each obstacle chosen from a uniform random distribution over the space $-5 \le y \le 5$, $0 \le y \le 10$. The side length for each obstacle was chosen from a Gaussian random distribution with mean 1.5 and standard deviation 0.5, and the orientation was chosen from a uniform distribution between 0° and 360°. In order to avoid generating trivially infeasible maps, any obstacles centered at

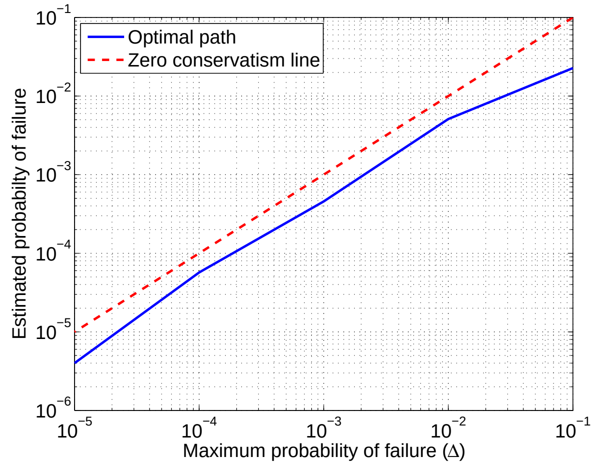

Fig. 12. Estimated probability of failure against maximum allowable probability of failure Δ. For comparison we show the line with unity gradient that corresponds to zero conservatism. The chance constrained plan is conservative by a factor of approximately 2.

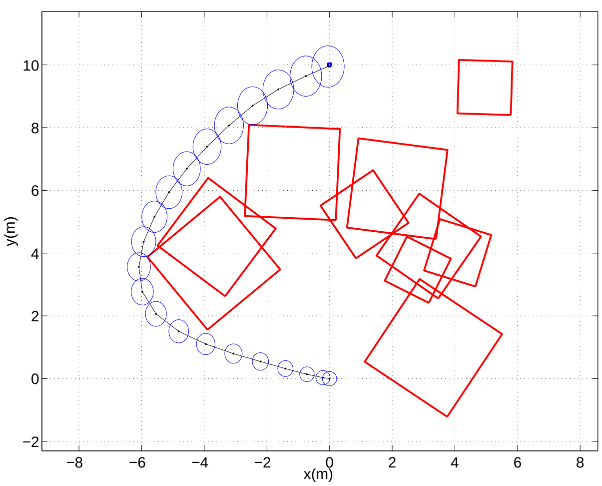

Fig. 13. Typical randomly generated map and path generated by customized solution method. The dots show the mean of the planned UAV position at each time step, and the ellipses give the 99% certainty region for the position.

a distance of 2.5 or less away from the goal location or the expected initial state of the UAV are removed and regenerated. For all examples in this section, $\Delta = 0.001$ and all other parameters are as in Section XI-A. A typical map, along with the solution returned by the chance constrained path planning algorithm, is shown in Figure 13. Table I summarizes the results from the randomized study. The suboptimality bound ρ was computed as:

$$\rho \triangleq \frac{J_{UB} - J_{LB}}{J_{UB}}. \tag{70}$$

When averaging this value across all cases, we removed instances where the algorithm returned the trivial cost bounds $[-\infty, \infty]$. Empirically we found that this happened in only 2.3% of cases. The main result from Table I is that in almost all cases, the algorithm returns nontrivial cost bounds, and when it does, the returned solution is known to be suboptimal by at most 2.4% (averaged over all cases) and by at most 21.6% (3-sigma). We conclude that for this UAV example, the customized solution approach returns a solution very close to the global optimum to the full disjunctive convex program posed in Problem 4.

Average cost	13.02
Standard deviation of cost	3.54
Fraction returning feasible solution	89.4%
Fraction returning nontrivial bounds	97.7%
Suboptimality bound (ρ)	mean = 2.4%, s.d. = 6.4%
MILP solution time	mean = 8.6s, s.d. = 11.8s
Convex program solution time	mean = 10.26s, s.d. = 13.21

TABLE I. Summary of study over randomly generated maps. We use s.d. to mean "standard deviation".

C. Comparison with Particle Control

In this section we compare the new approach with our previous approach[33], which we refer to as 'particle control'. Figure 14 shows a typical solution generated by the particle control approach for the hard map in Figure 8. For this case we set $\Delta = 0.01$, and used 100 particles. There are three important differences between the approach introduced in the present paper, and the particle control approach. First, the approaches use two different strategies for handling the chance constraint. The new approach bounds the chance constraint, which guarantees that any solution satisfies the constraint, but introduces conservatism. By contrast, particle control approximates the chance constraint, by ensuring that the fraction of particles violating the constraints is at most Δ. As the number of particles approaches infinity, the approximation becomes exact. For a sufficiently large number of particles, this means that, on average, the true probability of failure is close to Δ. Second, the new approach applies only to Gaussian distributions, while particle control applies to arbitrary distributions. This generality comes, however, at the cost of computational complexity. The particle control approach requires the solution of a MILP that scales with the number of particles, and the number of obstacles. Hence for large particle sets in nonconvex regions, the approach becomes intractable. Finally, the particle control approach is stochastic, in the sense that each time the algorithm is run, it will generate a slightly different plan. The new approach, by contrast, is deterministic.

For the hard map example, we generated 30 particle control solutions, with 100 particles and with $\Delta = 0.01$. CPLEX was used to solve the resulting MILP, with a maximum optimality gap of 1%. The average solution time was 413s. Using 10^6 Monte Carlo simulations, the average estimated probability of failure was 0.041, and the standard deviation was 0.025. This compares to the new approach, which solved the same problem in 10.78s with an estimated probability of failure of 0.0051. Note that, by increasing the number of particles used, the probability of failure achieved using particle control could be made to approach 0.01, but this would

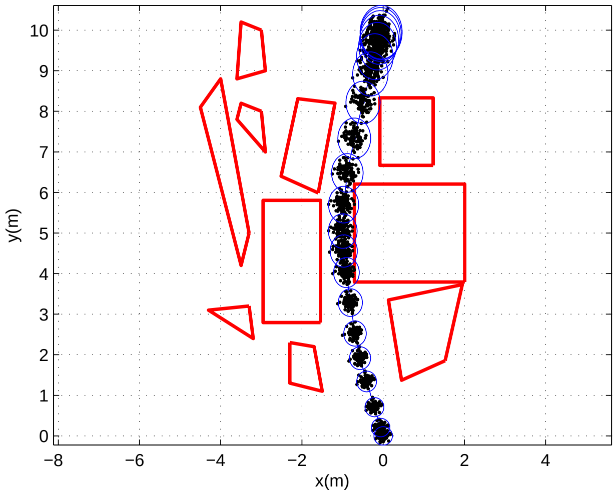

Fig. 14. Typical solution generated by particle control approach for the hard map, with $\Delta = 0.01$. The black dots show the 100 particles used in the plan. At most one particle can violate the constraints. The blue ellipses show the 99% certainty region for the position.

further increase computation time. Hence the new approach is significantly more computationally efficient than particle control and ensures that the chance constraint is satisfied.

XII. Conclusion

We have presented a new method for chance-constrained path planning with obstacles. The method optimizes a cost function, such as fuel, while ensuring that the probability of collision is below a user-specified threshold. Using a new bound on the probability of collision with the obstacles, the method approximates the nonconvex chance-constrained optimization problem as a Disjunctive Convex Program. This can be solved to global optimality using branch-and-bound techniques. We introduce a customized solution method that improves computation speed, and returns hard bounds on the level of suboptimality. We show empirically that, for a UAV path planning problem, the level of suboptimality is very small in almost all cases.

Acknowledgement

This research is funded in part by Boeing Company grant MIT-BA-GTA-1 and was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Government sponsorship acknowledged.

Appendix

A. Branch and Bound for Full Disjunctive Convex Program

Branch and bound for the general disjunctive convex program defined in Problem 2 works by exploring a search tree such as the one depicted in Figure 15. Each node of the tree corresponds to a convex subproblem, in which a subset of the conjunctive constraint clauses are imposed, and for each of the imposed conjunctive clauses one of the disjunctive clauses is imposed. The root node of the tree (Node 0) corresponds to the subproblem where only the equality constraints are imposed:

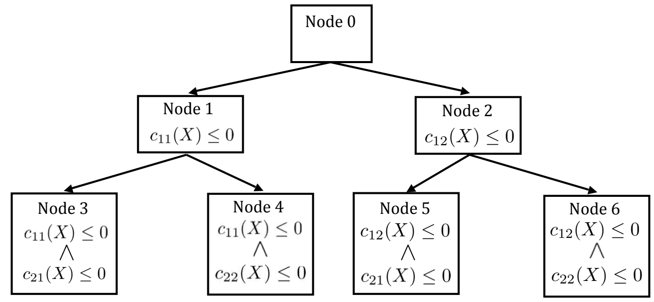

Fig. 15. Example search tree for general disjunctive convex program. In this case $n_{dis} = n_{cl} = 2$. Shown in the boxes are the inequality constraints applied at that node. Nodes are expanded to their children by branching on disjunctions and adding constraints to those imposed at the parent node. The equality constraints (not shown) are applied at all nodes.

Problem 8 (Root node subproblem).

$$\min_X h(X) \;\; \textit{subject to} \;\; f_{eq}(X) = \mathbf{0}. \tag{71}$$

Nodes in the tree are expanded by adding constraints to the subproblem. The main idea behind the branch and bound approach is that the optimal solution for a node is a lower bound on the cost of any subproblem below that node. Hence any node with an optimal cost greater than or equal to the best feasible solution found so far (the incumbent solution) need not be expanded. The tree is explored as follows:

1) *Initialize:* Set $J^*_{inc} = +\infty$ and $X^*_{inc} =$ *infeasible* and solve the root node subproblem. Add the root node to the stack.
2) *Node selection:* If the stack is empty, stop and return X^*_{inc}. Otherwise select a node i currently on the stack and solve its corresponding subproblem to find its optimal solution X^*_i and optimal cost J^*_i. Remove the node from the stack.
3) *Check constraints:* Check if the constraints in (10) are satisfied by X^*_i. If so, go to Step 4. If not, go to Step 5.
4) *Fathom check:* Check if $J^*_i \leq J^*_{inc}$. If so, set $X^*_{inc} = X^*_i$ and set $J^*_{inc} = J^*_i$. Go to Step 2.
5) *Node expansion:* Choose one disjunction in (10) not currently in subproblem i and add nodes to the tree for each of the clauses in the disjunction. Each new node adds the chosen clause to the conjunction of existing clauses. Go to Step 2.

An example of a subproblem, in this case corresponding to Node 5 in Figure 15, is:

Problem 9 (Node 5 subproblem).

$$\min_X h(X)$$

subject to:

$$\begin{aligned} & f_{eq}(X) = \mathbf{0} \\ & \Big(c_{12}(X) \leq 0\Big) \wedge \Big(c_{21}(X) \leq 0\Big). \end{aligned} \tag{72}$$

Since (72) is a conjunction of convex constraints, Problem 9 is a convex program. This is true of all the subproblems represented by nodes in the search tree. Since each subproblem is convex and can be solved to global optimality in finite time, and there are a finite number of possible subproblems that can be added to the tree, the branch and bound process is guaranteed to return the globally optimal solution to Problem 2 in finite time.

A great deal of research has been carried out that builds on the basic branch and bound algorithm to improve efficiency. Strategies include variable ordering, the use of *conflicts*, heuristics for node selection, and the addition of constraints to remove parts of the search tree. We do not claim to review this work here, but refer the interested reader to [44], [45], [7], [51].

B. Disjunctive Convex Programming using Binary Variables

The general Disjunctive Convex Program in Problem 2 can be encoded in an alternative form using binary variables. This encoding has been particularly popular in the special case of disjunctive linear programs, which become Mixed Integer Linear Programs (MILP) in this encoding. The MILP encoding allows the use of a great deal of performance-improving results in the literature, as well as highly-optimized software dedicated to solving MILPs[6]. We do not intend to review this body of work here, but refer the interested reader to [7], [52]. Instead we show how the encoding is carried out in the case of a disjunctive convex program.

The following Mixed Integer Convex Program (MICP) is equivalent to the Disjunctive Convex Program in Problem 2:

Problem 10 (Binary Variable Encoding).

$$\min_X h(X)$$

subject to:

$$f_{eq}(X) = \mathbf{0}$$

$$c_{ij}(X) \leq M\Big(1 - z_{ij}\Big) \ \ \forall i, j \tag{73}$$

$$\sum_{j=1}^{n_{cl}} z_{ij} \geq 1 \ \ \forall i \tag{74}$$

$$z_{ij} \in \{0, 1\} \ \ \forall i, j, \tag{75}$$

where M is a large positive constant. The binary variables in Problem 10 determine whether a particular constraint is imposed or not, while (73) ensures that at least one constraint in each disjunction is imposed, as required.

REFERENCES

[1] T. Schouwenaars, B. Mettler, E. Feron, and J. How, "Hybrid model for trajectory planning of agile autonomous vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, pp. 629–651, 2004.

[2] D. Jia and J. Vagners, "Parallel evolutionary algorithms for UAV path planning," in *Proc. AIAA 1st Intelligent Systems Technical Conference*, 2004.

[3] A. Richards and J. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. American Control Conference*, 2002.

[4] A. Chaudhry, K. Misovec, and R. D'Andrea, "Low observability path planning for an unmanned air vehicle using mixed integer linear programming," in *Proc. 43rd IEEE Conference on Decision and Control*, 2004.

[5] T. Schouwenaars, B. D. Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. European Control Conference*, 2001.

[6] ILOG, "ILOG CPLEX user's guide," 1999.

[7] C. A. Floudas, *Nonlinear and Mixed-Integer Programming - Fundamentals and Applications*. Oxford University Press, 1995.

[8] A. Richards, J. How, T. Schouwenaars, and E. Feron, "Plume avoidance maneuver planning using mixed integer linear programming," in *Proc. AIAA Guidance, Navigational and Control Conference*, 2001.

[9] A. Richards, "Robust constrained model predictive control," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

[10] A. Prékopa, *Stochastic Programming*. Kluwer, 1995.

[11] N. M. Barr, D. Gangsaas, and D. R. Schaeffer, "Wind models for flight simulator certification of landing and approach guidance and control systems," Paper FAA-RD-74-206, 1974.

[12] P. Li, M. Wendt, and G. Wozny, "A probabilistically constrained model predictive controller," *Automatica*, vol. 38, pp. 1171–1176, 2002.

[13] D. H. V. Hessem, "Stochastic inequality constrained closed-loop model predictive control," Ph.D. dissertation, Technische Universiteit Delft, Delft, The Netherlands, 2004.

[14] I. Batina, "Model predictive control for stochastic systems by randomized algorithms," Ph.D. dissertation, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2004.

[15] A. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AlChE Journal*, vol. 45, pp. 1743–1752, 1999.

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

[17] M. Ono and B. C. Williams, "An efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure," in *Proceedings of the Conference on Decision and Control*, 2008.

[18] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2009.

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

[20] V. Kunchev, L. Jain, V. Invancevic, and A. Finn, "Path planning and obstacle avoidance for autonomous mobile robots: A review," in *Knowledge-Based Intelligent Information and Engineering Systems*, ser. Lecture Notes in Computer Science, B. Gabrys, R. J. Howlett, and L. C. Jain, Eds. Springer Verlag, 2006, vol. 4252, pp. 537–544.

[21] J. Latombe, *Robot Motion Planning*. Kluwer Academic Press, 1991.

[22] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[23] T. Léaute and B. C. Williams, "Coordinating agile systems through the model-based execution of temporal plans," in *Proc. 20th National Conference on Artificial Intelligence*, 2005.

[24] J. Gossner, B. Kouvaritakis, and J. Rossiter, "Stable generalized predictive control with constraints and bounded disturbances," *Automatica*, vol. 33, p. 551, 1997.

[25] A. B. Acikmese and J. M. Carson, "A nonlinear model predictive control algorithm with proven robustness and stability," in *Proceedings of the American Control Conference*, June 2006, pp. 887–893.

[26] S. V. Raković and D. Q. Mayne, "A simple tube controller for efficient robust model predictive control of constrained linear discrete-time systems subject to bounded disturbances," in *Proceedings of the 16th IFAC World Congress*, 2005.

[27] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained Model Predictive Control using Linear Matrix Inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[28] A. A. Jalali and V. Nadimi, "A survey on robust model predictive control from 1999-2006," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, 2006.

[29] G. C. Calafiore and L. El Ghaoui, "Linear programming with probability constraints - part 2," in *Proceedings of the American Control Conference*, 2007.

[30] D. van Hessem, C. W. Scherer, and O. H. Bosgra, "LMI-based closed-loop economic optimization of stochastic process operation under state and input constraints," in *Proceedings of the Control and Decision Conference*, 2001.

[31] D. van Hessem and O. H. Bosgra, "Closed-loop stochastic dynamics process optimization under state and input constraints," in *Proceedings of the Control and Decision Conference*, 2002.

[32] ——, "A full solution to the constrained stochastic closed-loop MPC problem via state and innovations feedback and its receding horizon implementation," in *Proceedings of the Control and Decision Conference*, 2003.

[33] L. Blackmore, M. Ono, and A. Bektassov, "A probabilistic particle control approximation of chance constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, June 2010.

[34] A. Lambert and N. L. Fort-Piat, "Safe task planning integrating uncertainties and local maps federation," *International Journal of Robotics Research*, vol. 19, no. 6, 2000.

[35] A. Lambert and D. Gruyer, "Safe path planning in an uncertain-configuration space," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003.

[36] R. Pepy and A. Lambert, "Safe path planning in an uncertain-configuration space using RRT," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[37] L. Blackmore, H. X. Li, and B. C. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *Proceedings of the American Control Conference*, 2006.

[38] M. Ono and L. Blackmore, "Chance constrained finite horizon optimal control with nonconvex constraints," in *Proceedings of the American Control Conference*, 2010.

[39] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.

[40] L. Blackmore, A. Bektassov, M. Ono, and B. C. Williams, "Robust, optimal predictive control of jump markov linear systems using particles," in *Hybrid Systems: Computation and Control, HSCC*, ser. Lecture Notes in Computer Science, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. Springer Verlag, 2007, vol. 4416, pp. 104–117.

[41] L. Ljung, *System Identification: Theory for the user*. New York: Prentice Hall, 1999.

[42] Y. Ye, *Interior Point Algorithms*. Wiley, 1997.

[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[44] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Research*, vol. 14, pp. 699–719, 1966.

[45] R. E. Moore, "Global optimization to prescribed accuracy," *Computers and Mathematics with Applications*, vol. 21, no. 6/7, pp. 25–39, 1991.

[46] A. Prékopa, "Probabilistic programming," in *Handbooks in Operations Research and Management Science*, A. Ruszczyński and A. Shapiro, Eds. Elsevier Science, 2003, vol. 10, pp. 267–351.

[47] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, 2004.

[48] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, no. 1, 2005.

[49] L. Blackmore, "Robust path planning and feedback design under stochastic uncertainty," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2008.

[50] M. C. Campi and G. C. Calafiore, "New results on the scenario design approach," in *Proceedings of the Conference on Decision and Control*, 2007.

[51] H. Li and B. C. Williams, "Generalized conflict learning for hybrid discrete linear optimization," in *Proc. 11th International Conf. on Principles and Practice of Constraint Programming*, 2005.

[52] E. L. Johnson, G. L. Nemhauser, and M. W. P. Savelsbergh, "Progress in linear programming based branch-and-bound algorithms: Exposition," *INFORMS Journal of Computing*, vol. 12, 2000.