Chance Constrained Finite Horizon Optimal Control with Nonconvex Constraints

Masahiro Ono, Lars Blackmore, and Brian C. Williams

***Abstract*— This paper considers finite-horizon optimal control for dynamic systems subject to additive Gaussian-distributed stochastic disturbance and a chance constraint on the system state defined on a non-convex feasible space. The chance constraint requires that the probability of constraint violation is below a user-specified risk bound. A great deal of recent work has studied *joint* chance constraints, which are defined on the a conjunction of linear state constraints. These constraints can handle convex feasible regions, but do not extend readily to problems with non-convex state spaces, such as path planning with obstacles.**

In this paper we extend our prior work on chance constrained control in non-convex feasible regions to develop a new algorithm that solves the chance constrained control problem with very little conservatism compared to prior approaches.

In order to address the non-convex chance constrained optimization problem, we present two innovative ideas in this paper. First, we develop a new bounding method to obtain a set of decomposed chance constraints that is a sufficient condition of the original chance constraint. The decomposition of the chance constraint enables its efficient evaluation, as well as the application of the branch and bound method. However, the slow computation of the branch and bound algorithm prevents practical applications. This issue is addressed by our second innovation called Fixed Risk Relaxation (FRR), which efficiently gives a tight lower bound to the convex chance-constrained optimization problem. Our empirical results show that the FRR typically makes branch and bound algorithm 10-20 times faster. In addition we show that the new algorithm is significantly less conservative than the existing approach.

I. Introduction

Notation: The following notation is used throughout this paper.

$\boldsymbol{x}_t$: State vector at t'th time step (random variable)

$\boldsymbol{u}_t$: Control input at t'th time step.

$\boldsymbol{w}_t$: Additive disturbance at t'th time step (random variable)

$\bar{\boldsymbol{x}}_t$:= $E[\boldsymbol{x}_t]$: Nominal state at t'th time step

$$\boldsymbol{X} := \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix} \quad \boldsymbol{U} := \begin{bmatrix} \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{T-1} \end{bmatrix} \quad \bar{\boldsymbol{X}} := \begin{bmatrix} \bar{\boldsymbol{x}}_0 \\ \vdots \\ \bar{\boldsymbol{x}}_T \end{bmatrix}.$$

This research is funded 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.

Masahiro Ono is a PhD student, MIT. hiro_ono@mit.edu

Lars Blackmore is a Staff Engineer at the Jet Propulsion Laboratory, California Institute of Technology. Lars@jpl.nasa.gov

Brian C. Williams is a professor, MIT. williams@mit.edu

A. Overview and Problem Statement

This paper considers the problem of finite-horizon robust optimal control of dynamic systems under unbounded Gaussian-distributed uncertainty, with state and control constraints. We assume a discrete-time, continuous-state linear dynamics model. Gaussian-distributed stochastic uncertainty is a more natural model for exogenous disturbances such as wind gusts and turbulence[1], than the previously studied set-bounded models[2][3][4][5]. However, with stochastic uncertainty, it is often impossible to guarantee that state constraints are satisfied, since there is typically a non-zero probability of having a disturbance that is large enough to push the state out of the feasible region.

An effective framework to address robustness with stochastic uncertainty is optimization with *chance constraints*. Chance constraints require that the probability of violating the state constraints (i.e. the probability of failure) is below a user-specified bound known as the *risk bound*. An example problem is to drive a car to an destination as fast as possible while limiting the probability of an accident to 10^{-7}. This framework allows users to trade conservatism against performance by choosing the risk bound. The more risk the user accepts, the better performance they can expect.

Previous work [6][7][8][9][10] studied a specific form of chance constraint called a *joint chance constraint*, which is defined on the *conjunction* of linear state constraints. In other words, a joint chance constraint requires that the probability of satisfying *all* state constraints is more than $1-\Delta$, where Δ is the risk bound. Below is an example of a joint chance constraint defined on a conjunction of linear state constraints.

$$\Pr\left[\bigwedge_{i=1}^{N} \boldsymbol{h}_i^T \boldsymbol{X} \le g_i\right] \ge 1 - \Delta \qquad (1)$$

Here $\boldsymbol{h}$ is a vector, and the superscript T means transposition.

A clear limitation of the formulation in (1) is that the feasible state space needs to be convex, since a conjunction of linear state constraints defines a convex polytopic state constraint. However, many real-world problems have a non-convex state space. For example, the feasible region of a vehicle path planning problem with obstacles is non-convex. Another example is when a vehicle has to choose at which time step to pass through a particular region.

The objective of this paper is to solve an optimization problem with chance constraints defined over non-convex feasible regions, instead of the convex joint chance constraint in (1). An example of such a non-convex chance constraint is given in (2). Note that it contains disjunctions as well as conjunctions. The formal definition of non-convex chance constraints is given in the next subsection.

$$\Pr\left[((C_{\{1\}} \vee C_{\{2\}}) \wedge C_{\{3\}}) \vee C_{\{4\}}\right] \geq 1 - \Delta \tag{2}$$

where $C_{\{i\}}$ is the linear state constraint given by $h_i^T \boldsymbol{X} \leq g_i$.

There are two difficulties in handling the non-convex chance constraint. First, evaluating the chance constraint requires the computation of an integral over 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. 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 novel method to decompose the non-convex chance constraint 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.

While this approach is guaranteed to terminate in finite time, branch and bound requires many subproblems to be solved before the global optimum is found. Since in our case the convex subproblems are non-linear programs, this means that the overall computation time can be large. This problem is addressed by our innovation, called Fixed Risk Relaxation (FRR), which efficiently gives a tight lower bound to each convex chance-constrained optimization problem. The FRR is typically a linear or quadratic program, which can be solved efficiently. Using the bound from FRR we can more effectively prune the search space of the branch and bound approach, and our empirical results show that this yields a factor of 10-20 improvement in computation time.

B. Problem Statement

The open-loop finite-horizon optimal control problem with a non-convex chance constraint is formally stated as follows, where we assume that J is a proper, convex function:

$$\min_{\boldsymbol{U}} \quad J(\bar{\boldsymbol{X}}, \boldsymbol{U}) \tag{3}$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{w}_t \tag{4}$$

$$\boldsymbol{u}_{\min} \leq \boldsymbol{u}_t \leq \boldsymbol{u}_{\max} \tag{5}$$

$$\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \Sigma_w) \tag{6}$$

$$\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \Sigma_{x,0}) \tag{7}$$

$$\Pr\left[C_{\{\phi\}}\right] \geq 1 - \Delta \tag{8}$$

for all $t = 0, 1, \cdots T$. Eq.(8) is the general form of chance constraint that allows non-convex state constraints, and $C_{\{\phi\}}$ represents a possible non-convex set of state constraints.

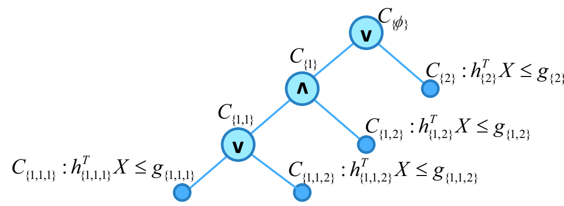

Fig. 1. The tree structure of the example chance constraint (10)

The set of state constraints $C_{\{i\}}$ is defined *recursively* by the following equation. It is either a linear state constraint, a conjunctive clause of state constraints, or a disjunctive clause of state constraints:

$$C_{\{i\}} := \begin{cases} \boldsymbol{h}_{\{i\}}^T \boldsymbol{X} \leq g_{\{i\}} & : \text{linear constraint} \\ \bigwedge_j C_{\{i,j\}} & : \text{conjunctive clause} \\ \bigvee_j C_{\{i,j\}} & : \text{disjunctive clause} \end{cases} \tag{9}$$

where subscript i is a *set* of indexes. The root state constraint $C_{\{\phi\}}$ in (8) has an empty index ϕ, and the children clauses of $C_{\{i\}}$ have an additional index j. For example, consider the following non-convex chance constraint:

$$\Pr\left[((C_{\{1,1,1\}} \vee C_{\{1,1,2\}}) \wedge C_{\{1,2\}}) \vee C_{\{2\}}\right] \geq 1 - \Delta. \tag{10}$$

The state constraints in the non-convex chance constraint in (10) have the following structure.

$$C_{\{\phi\}} := C_1 \vee C_2, \quad C_1 := C_{\{1,1\}} \wedge C_{\{1,2\}}$$
$$C_{\{1,1\}} := C_{\{1,1,1\}} \vee C_{\{1,1,2\}}$$

Intuitively, a set of state constraints can be represented as a tree. The example state constraints in the non-convex chance constraint in (10) have the tree shown in Figure 1.

C. Non-convex Chance Constraints

This subsection describes two example path planning problems, in order to illustrate the need for non-convex chance constraints.

1) Obstacle Avoidance: When planning a path in an environment with an obstacle such as Fig. 2-Left, the non-convex feasible region is approximated by the disjunction of linear constraints. The probability of penetrating into the obstacle should be limited to the given bound Δ at all time steps within the planning horizon $1 \leq t \leq T$. In such a case the chance constraint contains a disjunction of linear constraints as follows:

$$\Pr\left[\bigwedge_{t=1}^{T} \bigvee_{i=1}^{4} \boldsymbol{h}_{\{t,i\}}^T \boldsymbol{X} \leq g_{\{t,i\}}\right] \geq 1 - \Delta \tag{11}$$

2) Going Through a Region (Waypoint): When planning a path that goes through a region such as Fig. 2-Right, the set of constraints only has to be satisfied at one time step during the planning horizon. Therefore, the corresponding chance constraint is defined on the disjunction of the set of linear constraints as follows:

$$\Pr\left[\bigvee_{t=1}^{T} \bigwedge_{i=1}^{4} \boldsymbol{h}_{\{t,i\}}^T \boldsymbol{X} \leq g_{\{t,i\}}\right] \geq 1 - \Delta \tag{12}$$

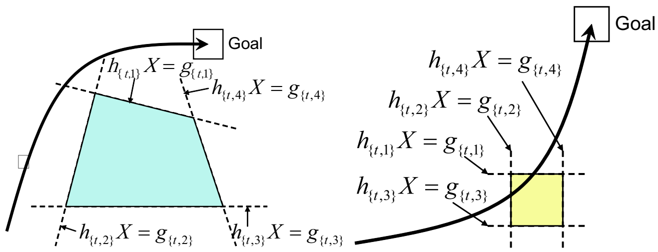

Fig. 2. Left: obstacle avoidance, Right: go-through constraint. In both cases, the chance constraint has disjunctive clauses of linear constraints.

3) General Path Planning Problem: Typically, a path planning problem has both obstacles and waypoints, and the resulting chance constraint has a complicated structure of conjunctive and disjunctive clauses.

D. Related Work

There is a significant body of work that solves *convex* joint chance constrained optimizations, many of which have been proposed in the context of model predictive control (MPC). MPC is a closed-loop control approach that, at each time step, solves a finite-horizon optimal control problem from the current initial state, executes the first step in the resulting optimal control sequence, and then resolves at the next time step. We are not concerned with such *receding-horizon* approaches in the present paper, and to the authors' knowledge, no results exist that guarantee the satisfaction of chance constraints for a closed-loop receding-horizon scheme. For early results in this area, we refer the reader to [11]. However there are a number of results in the MPC literature that address the *finite-horizon* chance-constrained optimal control problem. In the case of Gaussian uncertainty distributions, linear system dynamics and convex feasible regions, [12] considered chance constraints on individual scalar values. The extension from scalar random variables to joint random variables is essential if we wish to constrain the probability of failure over the entire planning horizon. The work of [13], [14] considered chance constraints on joint random variables, using the result of [15] to show that the optimization resulting from the chance-constrained finite-horizon control problem is convex, and can therefore be solved effectively using standard nonlinear solvers. This approach is limited, however, by the need to evaluate the multivariate Gaussian integrals in the constraint functions. These integrals are approximated through sampling, which is time-consuming and leads to approximation error.

[16] used a conservative ellipsoidal set bounding approach to ensure that the chance constraints are satisfied without the need for the evaluation of multivariate Gaussian integrals. The key idea is to characterize a region around the state mean that the state is guaranteed to be in with a certain probability (the '99%' region) and ensure that this deterministic set satisfies the constraints. The approach of [16] explicitly optimized over feedback laws as well as feedforward controls[1]. An alternative bounding approach is to use Boole's inequality to split joint chance constraints over N variables into N univariate chance constraints, and to ensure that the probability of violation of each of these is at most δ/N, where δ is the specified maximum probability of failure. This approach was suggested by [8] and [17] for convex feasible regions. Another bounding approach was proposed by [6], which does not require that all uncertainty is Gaussian. In this work, the authors draw n samples or 'scenarios' from the random variables and ensure that the constraints are satisfied for all of the samples. The bound in the scenario approach is stochastic, in the sense that n is chosen to ensure that the chance constraint is satisfied with probability $1-\beta$, where β is small and chosen by the user. Bounding approaches are, however, prone to excessive conservatism whereby the true probability of constraint violation is far lower than the specified allowable level. Conservatism leads to excess cost and can prevent the optimization from finding a feasible solution at all.

Previous work proposed letting the risk of constraint violation be an explicit optimization parameter, rather than being fixed. We refer to this approach as *risk allocation*. Risk allocation was introduced in [17] for chance constrained linear programming, and was applied to finite horizon optimal control in [18], [10], [19]. [17] showed that for convex polytopic state constraints the problem can be solved as a single convex optimization problem, and in [19] we showed that the resulting conservatism is very low.

On the other hand, there are only two prior methods [9][20] that handle non-convex chance-constrained optimization, as far as the authors know. However, although the sampling based method [9] is theoretically applicable to any chance-constraints including non-convex ones, its slow computation prevents its practical use. The other method [20] fixes the individual risk bounds, and solves the resulting mixed-integer linear programming. Although the approach is efficient, the fixed risk bounds introduces unnecessary conservatism. In the present paper we introduce a new algorithm that uses the risk allocation approach to avoid excessive conservatism, while still allowing for efficient computation.

II. Decomposition of General Chance Constraint

There are two difficulties to handle the chance constraint defined by (8),(9). First, it is very hard to evaluate due to the difficulty of computing an integral of multi-variable probability distribution over an arbitrary region. Second, the branch and bound method, which is a standard approach to non-convex optimization, cannot be directly applied, since the chance constraint (8) is one single constraint that cannot be divided. Disjunctions (i.e. non-convexity) only appear inside of the chance constraint.

Our approach to address these two issues is to decompose conjunctive and disjunctive clauses of a chance constraint into a set of *individual chance constraints*, which are defined

[1] Note however, that the chance constraints are only guaranteed to hold in finite-horizon (open loop) execution, rather than in receding horizon (closed loop).

on uni-variable probability distributions. In other words, we move the conjunctions and disjunctions out of probability. The resulting set of individual chance constraints are a sufficient condition of the original chance constraint.

We decompose the chance constraint (8)(9) recursively, one by one, from top to the bottom of the tree shown in Fig. 1. Different rules are used to decompose conjunctive and disjunctive clauses.

A. Decomposition of Conjunctive Clause: Risk Allocation

In this subsection we consider a conjunctive clause of chance constraints:

$$\Pr\left[\bigwedge_{i=1}^{N} C_i\right] \geq 1-\Delta \tag{13}$$

where C_i is a linear state constraint, or a set of linear state constraints. Our approach is to obtain a decomposed form of chance constraint that is a sufficient condition of (13) using the union bound or Boole's inequality:

$$Pr[A \cup B] \leq \Pr[A] + \Pr[B]. \tag{14}$$

Observe that, using the Boole's inequality, the conjunctive joint chance constraint (13) is implied by the following conditions.

$$\bigwedge_{i=1}^{N} \left(\Pr[C_i] \quad \geq \quad 1-\delta_i\right) \tag{15}$$

$$\forall_i \quad \delta_i \quad \geq \quad 0 \tag{16}$$

$$\sum_{i=1}^{N} \delta_i \quad \leq \quad \Delta \tag{17}$$

Note that a chance constraint defined on a conjunction of state constraints (13) is decomposed to a conjunction of chance constraints defined on individual state constraints (15). Each chance constraint in (15) has its own risk bound δ_i. Eq.(16) is necessary since δ_i are probabilities. Eq.(17) says that the sum of δ_i is upper-bounded by the original risk bound Δ. Past work [8] and [20] fixed $\delta_1 \cdots \delta_N$ to arbitrary values, such as $\delta_i = \Delta/N$. We treat them as decision variables that are optimized along with $\boldsymbol{U}$.

The optimization problem of $\delta_1 \cdots \delta_N$ can be viewed as a resource allocation problem; each chance constraint is assigned resource (risk) δ_i, whose total amount is limited. The goal of the optimization problem is to find the best allocation of the resource $\delta_1^\star \cdots \delta_N^\star$ that minimizes the cost. Thus we call $\delta_1^\star \cdots \delta_N^\star$ a "*risk allocation*". Methods for optimizing the risk allocation in the case of a single conjunctive clause were introduced in [10], [17], [19]. We extend this to handle an arbitrary combination of conjunctive and disjunctive clauses in this paper.

B. Decomposition of Disjunctive Clause: Risk Selection

In this subsection we consider a disjunctive clause of chance constraints:

$$\Pr\left[\bigvee_{i=1}^{N} C_i\right] \geq 1-\Delta \tag{18}$$

where C_i is a linear state constraint, or a set of linear state constraints.

The following inequalities always hold:

$$\forall_i \quad \Pr\left[\bigvee_{i=1}^{N} C_i\right] \geq \Pr\left[C_i\right] \tag{19}$$

Therefore, (18) is implied by the following:

$$\bigvee_{i=1}^{N} \left(\Pr[C_i] \geq 1-\Delta\right) \tag{20}$$

Note that a chance constraint defined on a disjunction of state constraints (18) is decomposed to a disjunction of chance constraints defined on individual state constraints (20). All the individual chance constraints in (20) has the same risk bound as the original chance constraint Δ.

C. Recursive Decomposition for General Clause

We now show that the two decomposition rules can be applied recursively to decompose a general clause with both disjunctive clauses and conjunctive clauses, resulting in chance constraints on individual state constraints only. For example, the chance constraint in the example (10) is decomposed to individual chance constraints as follows:

$$\begin{aligned}
& \Pr\left[\left((C_{\{1,1,1\}} \vee C_{\{1,1,2\}}) \wedge C_{\{1,2\}}\right) \vee C_{\{2\}}\right] \geq 1-\Delta \\
\Leftarrow\ & \Pr\left[(C_{\{1,1,1\}} \vee C_{\{1,1,2\}}) \wedge C_{\{1,2\}}\right] \geq 1-\Delta \\
& \quad \vee \quad \Pr\left[C_{\{2\}}\right] \geq 1-\Delta \\
\Leftarrow\ & \{\Pr\left[(C_{\{1,1,1\}} \vee C_{\{1,1,2\}})\right] \geq 1-\delta_1 \\
& \quad \wedge \quad \Pr\left[C_{\{1,2\}}\right] \geq 1-\delta_2 \\
& \quad \wedge \quad \delta_1+\delta_2 \leq \Delta\} \quad \vee \quad \Pr\left[C_{\{2\}}\right] \geq 1-\Delta \\
\Leftarrow\ & \{(\Pr[C_{\{1,1,1\}}] \geq 1-\delta_1 \vee \Pr[C_{\{1,1,2\}}] \geq 1-\delta_1) \\
& \quad \wedge \quad \Pr[C_{\{1,2\}}] \geq 1-\delta_2 \quad \wedge \quad \delta_1+\delta_2 \leq \Delta\} \\
& \quad \vee \quad \Pr[C_{\{2\}}] \geq 1-\Delta
\end{aligned} \tag{21}$$

Note that this decomposition introduces conservatism due to the difference between the left hand side and the right hand side of the inequalities (14) and (19). However, we claim that this suboptimality is much less than previous bounding approaches, such as [8], [16], [20]. We showed empirically that this is the case for convex constraints in [10] and [19]. In Section V we show that this is also the case for general non-convex constraints.

III. Branch and Bound Algorithm

We solve the optimization problem (3)-(7) with the decomposed chance constraints. That is, instead of (10) we use the decomposition in (21) to give:

$$\begin{aligned}
& \{(\Pr[C_{\{1,1,1\}}] \geq 1-\delta_1 \vee \Pr[C_{\{1,1,2\}}] \geq 1-\delta_1) \\
& \quad \wedge \quad \Pr[C_{\{1,2\}}] \geq 1-\delta_2 \quad \wedge \quad \delta_1+\delta_2 \leq \Delta\} \\
& \quad \vee \quad \Pr[C_{\{2\}}] \geq 1-\Delta
\end{aligned} \tag{22}$$

This is a non-convex optimization problem. Its optimal solution is found by the following process. First, we find all possible *conjunctive* combinations of constraints by choosing one set of constraints at each disjunction. In the example of (22), we can find three conjunctive combinations as follows:

$$\begin{aligned}&\Pr[C_{\{1,1,1\}}] \geq 1-\delta_1 \wedge \Pr[C_{\{1,2\}}] \geq 1-\delta_2 \wedge \delta_1+\delta_2 \leq \Delta \\ &\Pr[C_{\{1,1,2\}}] \geq 1-\delta_1 \wedge \Pr[C_{\{1,2\}}] \geq 1-\delta_2 \wedge \delta_1+\delta_2 \leq \Delta \\ &\Pr[C_{\{2\}}] \geq 1-\Delta \end{aligned} \tag{23}$$

A candidate solution is feasible for the original non-convex chance constraint (10) if it is feasible for *any* of the conjunctive combinations in (23). In [19] we showed that each of the conjunctive combinations, in general, yields a set of convex constraints. We therefore propose solving the original non-convex optimization problem (3)-(7) by searching over the conjunctive combinations, solving for each conjunctive combination a convex optimization using the approach of [19] and returning the best solution from all of the convex optimizations. Since convex optimization problems can be solved to global optimality using existing solvers, as long as the decomposition in Section II does not introduce too much conservatism, this approach will return solutions close to the global optimum of the original non-convex chance constrained problem. The drawback of this approach is that non-convex problems often have large numbers of disjunctions, which lead to a large numbers of conjunctive combinations and hence many convex optimizations to be solved. This will lead to large computation times.

To overcome this problem, we avoid having to solve a convex program for every possible conjunctive combination using the branch and bound algorithm. Fig. 3 shows the search tree used by the branch and bound algorithm for the example problem with the non-convex chance constraint (22). The leaf nodes (represented by squares in Fig. 3) represent all conjunctive combination of state constraints, while branching nodes (represented by circles) correspond to disjunctions of the non-convex chance constraint. At each leaf node of the search tree, the algorithm solves a convex joint chance constrained problem using the approach of [19]. A relaxed problem is solved at each branching node in order to give the lower bound of all leaf nodes below the branching node (see Fig. 3). The algorithm searches for the best solution in a depth-first manner, and if the solution of a relaxed problem at a branching node is worse than the current best solution, the branch is pruned. This process ensures that the globally optimal solution is found, and typically ensures that a small subset of the nodes are evaluated.

We propose a bounding approach whereby the relaxed problems are constructed by removing all constraints below the corresponding disjunction. This approach was used by [21] and [22] for a different problem known as disjunctive linear programming. For example, the relaxed problem at the middle left node in Fig. 3 is constructed by removing the first two constraints $\Pr[C_{\{1,1,1\}}] \geq 1-\delta_1$ and $\Pr[C_{\{1,1,2\}}] \geq 1-\delta_1$. The resulting relaxed problems are also convex joint chance constrained optimization problems. In this paper we use this approach to perform bounding, and present results in Section V. The drawback of this bounding approach is that, while the number of nodes evaluated is reduced, each

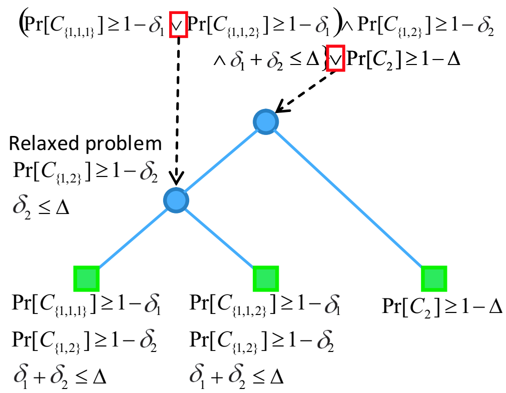

Fig. 3. The search tree of the branch and bound algorithm for the decomposed non-convex chance constraints (22)

node evaluation requires a *non-linear* convex program to be solved. Solution time for a convex non-linear program is on the order of seconds for problems with 50 state constraints, which means that computation time is a serious issue. In the next section we therefore propose an additional bounding approach that does not require solution of a non-linear program, and dramatically reduces computation times.

IV. Fixed Risk Relaxation

In this section we formulate a relaxed optimization problem, namely Fixed Risk Relaxation (FRR), to efficiently obtain a lower bound of convex chance-constrained optimization problems. FRR is used at both leaf nodes and branching nodes; at the branching nodes, we solve the FRR of the relaxed problem described in the previous section, instead of the relaxed problem itself.

The FRR only has linear constraints. Therefore, when the objective function (3) is linear or quadratic, which is the case for many applications, the optimization problem with FRR are linear/quadratic programs, which can be solved efficiently.

A. Linearization of Individual Chance Constraints

The only non-linear constraints in the convex chance-constrained optimization problem are the individual chance constraints, such as the ones in (22). Below is an individual chance constraint, which is defined on a single linear state constraint:

$$\Pr[\boldsymbol{h}_i^T \boldsymbol{X} \leq g_i] \geq 1-\delta_i$$

The individual chance constraint, which is defined on a random variable $\boldsymbol{X}$, is equivalent to deterministic constraints defined on the nominal state $\bar{\boldsymbol{X}}$ as follows[23]:

$$\boldsymbol{h}_i^T \bar{\boldsymbol{X}} \leq g_i - m_i(\delta_i) \tag{24}$$

where $-m_i(\cdot)$ is the inverse of cumulative distribution function of univariable Gaussian distribution with variance $\boldsymbol{h}_i^T \Sigma_X \boldsymbol{h}_i$. Note the negative sign.

$$m_i(\delta_i) = -\sqrt{2\boldsymbol{h}_i^T \Sigma_X \boldsymbol{h}_i}\ \mathrm{erf}^{-1}(2\delta_i - 1) \tag{25}$$

where erf^{-1} is the inverse of the Gauss error function and Σ_X is the covariance matrix of $\boldsymbol{X}$.

The deterministic form of chance constraint (24) is non-linear due to the inverse cumulative distribution function $-m_i(\cdot)$. Our idea is to turn these non-linear constraints into linear constraints by fixing δ_i, and hence, making $m_i(\delta_i)$ a constant.

B. Fixed Risk Relaxation

The fixed risk relaxation (FRR) of a chance-constrained optimization problem is obtained by fixing all individual risk bounds δ_i to the original risk bound Δ:

$$\forall_i \quad \delta_i = \Delta \tag{26}$$

Lemma 1: The optimization problem (3)-(7) with the Fixed Risk Relaxation gives a lower bound on the cost of the original convex chance-constrained optimization problem.

Proof: It immediately follows from (16) and (17) that $\forall_i \quad \delta_i \leq \Delta$. Since $m_i(\cdot)$ is a monotonically increasing function, all individual chance constraints (24) of the Fixed Risk Relaxation are looser than the original problem. Therefore, the cost of the optimal solution of the Fixed Risk Relaxation is less than or equal to the original problem. ■

Note that the solution of the optimization problem with FRR is an infeasible solution to the original problem, since (26) violates the constraint (17). In a special case where there is only one individual chance constraint, such as the relaxed problem in Fig. 3, the Fixed Risk Relaxation is equivalent to the original problem.

C. Using FRR in Branch and Bound Algorithm

By using the FRR, the Branch and Bound algorithm can be substantially sped up. In each node of the Branch and Bound algorithm, the FRR is solved first. If the lower bound given by the FRR is more than the incumbent, the node is pruned without solving the original convex chance-constrained optimization; otherwise, the node is expanded at the branching nodes, and the original convex chance-constrained optimization is solved at the leaf nodes.

V. SIMULATIONS

A. Problem Settings

We tested our methods on two 2-D path planning problems: Obstacle Avoidance problem, and Go-Through-Waypoints problem. Fig. 4 shows the example results of the Obstacle Avoidance and the Go-Through-Waypoints problem. A vehicle starts from $[0,0]$, and heads to the rectangular goal region with its center at $[1.05, 1.05]$ and the edge length 0.1. In the Obstacle Avoidance problem, a rectangular obstacle with its edge length 0.6 is placed at a random location within the square region with its corners at $[0,0]$, $[1,0]$, $[1,1]$, and $[0,1]$. In the Go-Through-Waypoint problem, two rectangular waypoints (regions) with their edge length 0.1 are placed at random locations within the same square region. The risk bound is set to $\Delta = 0.001$ for both problems. The following discrete-time dynamics model is used.

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{B} = \begin{pmatrix} \Delta t^2/2m & 0 \\ 0 & \Delta t^2/2m \\ \Delta t/m & 0 \\ 0 & \Delta t/m \end{pmatrix}$$

$$\Delta t = 0.5, m = 1$$

$$u_{\min} = [-0.5, -0.5]\,, u_{\max} = [0.5, 0.5]$$

$\boldsymbol{w}_t$ is sampled from a zero-mean Gaussian distribution with variance.

$$\Sigma_w = \begin{pmatrix} 10^{-5} & 0 & 0 & 0 \\ 0 & 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The cost is the total control input during the planning horizon $1 \leq t \leq T$:

$$J(\bar{\boldsymbol{X}}, \boldsymbol{U}) = \sum_{t=1}^{T} \left(|u_x| + |u_y|\right).$$

B. Results

Fig. 4 shows the solutions given by the proposed algorithm. The circles represent three standard deviations of the distribution of vehicle locations, while the plus marks ('+') represent the nominal location at each time step. The resulting probabilities of constraint violation in the examples are $1 - \Pr\left[C_{\{\phi\}}\right] = 0.000938$ for the Obstacle Avoidance problem, and 0.000991 for the Go-Through-Waypoints problem, both of which satisfy the given chance constraint $1 - \Pr\left[C_{\{\phi\}}\right] \leq 0.001$. These results show that our proposed method successfully guides the vehicle to the goal while respecting the chance constraint in both problems. In Fig. 4-Top, it appears that the path cuts across the obstacle. This is due to the discretization of the plant dynamics; the optimization problem only requires that the vehicle locations at each discrete time step satisfies the constraints, and does not care about the state in between. This issue can be addressed by a constraint tightening method[24].

Table I compares the performance of three algorithms on the Obstacle Avoidance problems and the Go-Through-Waypoints problem. The three algorithms are Branch and Bound with optimized risk allocation and Fixed Risk Relaxation (FRR) (the proposed algorithm), the Branch and Bound with optimized risk allocation but without the FRR, and our previous method [20] that uses a fixed risk allocation. Although [20] only deals with obstacle avoidance problems, we have extended the approach here to Go-Through-Waypoints problems in order to be compared with the algorithm proposed in the present paper. The values in the table are the averages of 20 runs with random locations for the obstacle and waypoints. The probability of constraint violation ($1 - \Pr\left[C_{\{\phi\}}\right]$) is evaluated by Monte-Carlo simulation with 10^6 samples.

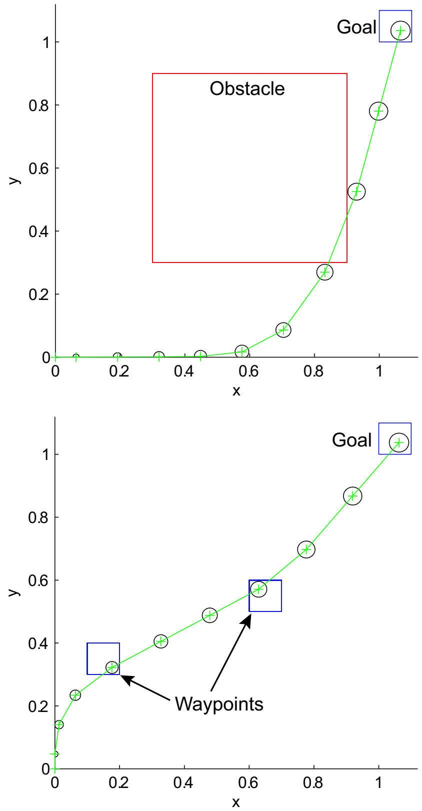

Fig. 4. Example simulation results. Top: Obstacle Avoidance problem, Bottom: Go-Through-Waypoints problem. The circles represent three standard deviations of the distribution of vehicle locations.

a) Conservatism: As discussed in Section II, satisfaction of the decomposed chance constraints is a sufficient condition for satisfaction of the original chance constraint (8). Conservatism is introduced by the difference between the two sides of the inequalities (14) and (19).

The results show that the conservatism of our proposed method is significantly smaller than the fixed risk allocation method. In almost all problems of interest, the chance constraint is active. This means that when the solution is exactly optimal, the probability of constraint violation $1 - \Pr\left[C_{\{\phi\}}\right]$ is equal to the risk bound Δ, which is set to $\Delta = 0.001$ in our simulations. Table I shows that the probability of constraint violation of our proposed algorithm is very close to Δ. On the other hand, the fixed risk allocation method [20] has non-negligible conservatism; its probability of constraint violation is less than the half of the given risk bound. This significant conservatism is due to the fixed individual risk bounds (risk allocation).

We cannot evaluate the suboptimality in terms of the cost function, since the exactly optimal solution is unavailable. Nonetheless, we can observe in Table I that the proposed approach results in a better cost than the fixed risk allocation method.

b) Computation time: The cost of reduced conservatism is the increased computation time. However, the results show that the FRR significantly enhances the computation speed of the Branch and Bound algorithm in both problems. As shown in Table I, although the algorithm with FRR always results in the exactly same solution as the one without FRR, its computation is 10-20 times faster. Note that the advantage of FRR is smaller on the Go-Through-Waypoints problem than on the Obstacle Avoidance problem. This is due to the shallow depth of the search tree. Typically, the advantage of using FRR is more significant in a problem with deep search tree such as the obstacle avoidance problem. Since a problem with a deep search tree typically requires larger computation time, it can be said that the advantage of using FRR is more significant in difficult problems.

TABLE I

COMPARISON OF COMPUTATION TIME, PROBABILITY OF CONSTRAINT VIOLATION, AND COST OF THREE ALGORITHMS. THE VALUES ARE THE AVERAGES OF 20 RUNS WITH RANDOM LOCATION OF OBSTACLE AND WAYPOINTS. THE SECOND ROW SHOWS THE RESULTING PROBABILITY OF CONSTRAINT VIOLATION. THE RISK BOUND IS SET TO $\Delta = 0.001$.

	Optimized risk allocation		Fixed risk allocation
	w/ FRR	w/o FRR	
Obstacle Avoidance problem			
Comp. time [sec]	35.97	875.38	2.56
PCV*	9.975×10^{-4}		2.829×10^{-4}
Cost	0.352		0.357
Go-Through-Waypoints problem			
Comp. time [sec]	25.53	283.32	0.656
PCV*	9.784×10^{-4}		4.061×10^{-4}
Cost	0.576		0.585

*PCV = Probability of constraint violation

VI. CONCLUSION

We proposed two innovative ideas to solve non-convex chance constrained optimization problem efficiently with small suboptimality. The first is the recursive decomposition technique of a chance constraint (Section II), which enables its efficient evaluation, as well as the application of the branch and bound method. The second is the Fixed Risk Relaxation (Section IV), which makes the branch and bound algorithm significantly faster by giving lower bounds to convex chance-constrained optimization problems efficiently. The validity and efficiency of our method was demonstrated by simulations.

ACKNOWLEDGMENTS

Thanks to Michael Kerstetter, Scott Smith and the Boeing Company for their support.

REFERENCES

[1] 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.

[2] E. C. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. dissertation, University of Cambridge, 2000.

[3] 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, October 1996.

[4] J. Löfberg, “Minimax approaches to robust model predictive control,” Ph.D. dissertation, Linköping Studies in Science and Technology, 2003.

[5] Y. Kuwata, A. Richards, and J. How, “Robust receding horizon control using generalized constraint tightening,” *Proceedings of American Control Conference*, 2007.

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

[7] D. H. van Hessem, “Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation,” Ph.D. dissertation, Delft University of Technology, 2004.

[8] A. Nemirovski and A. Shapiro, “Convex approximations of chance constrained programs,” *SIAM Journal on Optimization*, vol. 17, pp. 969–996, 2006.

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

[10] M. Ono and B. C. Williams, “Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint,” in *Proceedings of 47th IEEE Conference on Decision and Control (CDC-08)*, 2008.

[11] J. A. Primbs, “Receding horizon control of constrained linear systems with state and control multiplicative noise,” in *Proceedings of the American Control Conference*, 2007.

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

[13] P. Li, M. Wendt, and G. Wozny, “Robust model predictive control under chance constraints,” *Computers and Chemical Engineering*, vol. 24, pp. 829–834, 2000.

[14] ——, “A probabilistically constrained model predictive controller,” *Automatica*, vol. 38, pp. 1171–1176, 2002.

[15] P. Kall and S. W. Wallace, *Stochastic Programming*. Wiley, 1994.

[16] 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 *IEEE Conference on Decision and Control*, Orlando, FL, Dec. 2001.

[17] A. Prékopa, “The use of discrete moment bounds in probabilistic constrained stochastic programming models,” *Annals of Operations Research*, vol. 85, pp. 21–38, 1999.

[18] M. Ono and B. C. Williams, “Efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure,” in *Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-08)*, Chicago, IL, July 2008.

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

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

[21] E. Balas, “Disjunctive programming,” *Annals of Discrete Mathematics*, 1979.

[22] 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.

[23] A. Prekopa, *Stochastic Programming*. Kluwer, 1995.

[24] Y. Kuwata, “Real-time trajectory design for unmanned aerial vehicles using receding horizon control,” Master’s thesis, Massachusetts Institute of Technology, 2003.