Joint Chance-Constrained Dynamic Programming

Masahiro Ono, Yoshiaki Kuwata, and J. (Bob) Balaram

***Abstract*— This paper presents a novel dynamic programming algorithm with a *joint chance constraint*, which explicitly bounds the risk of failure in order to maintain the state within a specified feasible region. A joint chance constraint cannot be handled by existing constrained dynamic programming approaches since their application is limited to constraints in the same form as the cost function, that is, an expectation over a sum of one-stage costs. We overcome this challenge by reformulating the joint chance constraint into a constraint on an expectation over a sum of indicator functions, which can be incorporated into the cost function by dualizing the optimization problem. As a result, the primal variables can be optimized by a standard dynamic programming, while the dual variable is optimized by a root-finding algorithm that converges exponentially. Error bounds on the primal and dual objective values are rigorously derived. We demonstrate the algorithm on a path planning problem, as well as an optimal control problem for Mars entry, descent and landing. The simulations are conducted using a real terrain data of Mars, with four million discrete states at each time step.**

I. Introduction

When controlling a system with stochastic uncertainty, it is important to limit the risk of mission failure while minimizing a given cost function. For example, Mars entry, descent, and landing (EDL) is subject to various source of uncertainties such as atmospheric variability and imperfect aerodynamics model. The resulting dispersions of the landing position typically spans over tens of kilometers for a 99.9% confidence ellipse [1]. Given such a highly uncertain nature of EDL, a target landing site must be carefully chosen in order to limit the risk of landing on rocky or uneven terrain, since such an event may directly results in a mission failure. At the same time, it is equally important to land near science targets in order to minimize the traverse distance after the landing. An effective framework for such a multi-objective decision making involving risk can be posed as a chance-constrained optimization, a minimization of the expected cost with a bound on the risk of mission failure [2], [3].

Future Mars lander/rover missions aim to reduce the uncertainty by using several new active control technologies, consisting of the following three stages: entry-phase targeting, powered-descent guidance (PDG) [4], and hazard detection and avoidance (HDA) [5], as shown in Figure 1. Each control stage is capable of making corrections to the predicted landing position by a certain distance, but each stage is subject to execution errors, which deviates the spacecraft away from the planned landing position. We pose this problem as an optimal sequential decision making under a persisting uncertainty.

Masahiro Ono (ono@appi.keio.ac.jp) is with Keio University, Kanagawa, Japan. Yoshiaki Kuwata (Yoshiaki.Kuwata@jpl.nasa.gov) and J. (Bob) Balaram are with Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA, USA. Copyright 2012 California Institute of Technology. Government sponsorship acknowledged.

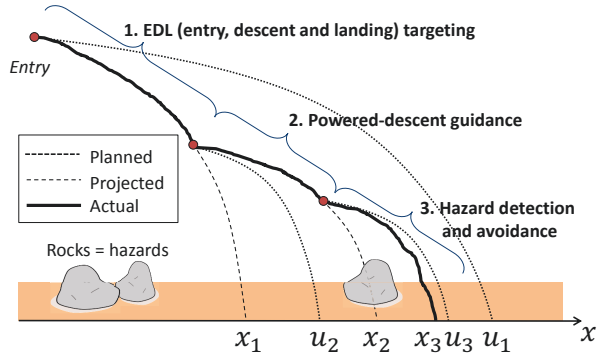


Fig. 1. A future Mars entry, descent, and landing scenario.

Dynamic programming (DP) is a general framework for optimal sequential decision making under uncertainty and provides theoretical basis for various optimal control methods, such as the linear-quadratic regulator (LQR) and the model-predictive control (MPC) [6]. Among the MPC community, the chance-constrained MPC (CCMPC) has been intensively studied over the last decade [7], [8], [9], [10], [11], [12], [13], [14] and has found successful applications such as control of buildings [15] and electrical grids [16]. The objective of this paper is to reflect the rich insights obtained from these CCMPC studies back to DP and develop a general joint chance-constrained dynamic programming framework that can be applied for a broad range of optimal control problems.

Chance-constrained dynamic programming was initially studied in the context of water management in the 1970s [17], [18]. Although these early studies are important in that they first introduced the notion of the chance constraint in DP, a field-specific formulation and a lack of theoretical justification of the employed optimization method are their weakness. A similar but different constraint called reliability constraint has been also studied [19], which limits the expected number of failures and employs a Lagrangian method to transform the constrained optimization into an unconstrained optimization problem. A closely related discipline is the Markov decision process (MDP), which typically achieves risk aversion by imposing an arbitrary penalty on failure states. The proposed chance-constrained DP approach is different from the penalty-based MDPs in that the former can explicitly impose a constraint on the probability of

failure, which is what the stake holders desire to have especially for space missions. Several algorithms have been proposed to solve constrained MDP problems [20]. Among them, the most relevant work is [21], which also employed a Lagrangian method to handle constraints. A limitation of the constrained MDP approaches is that constraint functions must have the same form as the objective function, that is, an expectation over a summation of one-stage costs. Due to this limitation, it is generally difficult to handle a chance constraint by constrained DP/MDP approaches, as pointed out by [19].

We overcome this challenge by using the recently-developed risk allocation approach [13], which decomposes a joint chance constraint into a set of individual chance constraints and distributes the risks among them. With this approach, a joint chance constraint is transformed into an expectation over a summation of indicator functions, which has the same form as the cost function. The resulting Lagrangian of the constrained optimization preserves the form that is handled by a standard DP. We propose an algorithm that solves the dual optimization problem, where the dual objective function is evaluated by minimizing the Lagrangian using a standard dynamic programming approach (i.e., backward recursion), given a fixed dual variable. The dual variable is optimized by a root-finding algorithm called Brent's method [22]. The root-finding algorithm has an exponential convergence rate, and its complexity does not change with the size of the primal problem. It typically converges within 10 to 30 iterations.

Another important contribution of this paper is the derivation of the bounds on the suboptimality of both the primal and dual objective values. The dual objective function of a chance-constrained optimization is often non-differentiable, and hence it is difficult to find an exact optimal solution with a root-finding algorithm. Our algorithm obtains an approximate solution, but with a user-specified tolerance on the suboptimality of the dual objective value. A suboptimality bound on the primal objective value is also obtained from the dual solution.

The rest of the paper is organized as follows. In Section II, the problem statement of chance-constrained DP is presented, together with the reformulation which allows us to apply a Lagrangian-based method. Section III then presents its dual problem, the proposed solution approach, and its suboptimality bounds. Section IV shows the algorithmic steps, and finally Section V shows several simulation results.

II. Formulation of Joint Chance-constrained DP

A. Problem Statement

We consider a discrete-time stochastic dynamic system, whose state at time k is represented by a vector $x_k \in \mathcal{X}$. The state space $\mathcal{X}$ can be continuous, discrete, or hybrid. We assume the following general dynamics model:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k) \\ u_k &\in \mathcal{X}_k(x_k) \\ w_k &\sim p_k(w_k), \quad (k = 0, \ldots, N-1) \end{aligned}$$

where u_k is a control input in a set $\mathcal{X}_k(x_k) \subseteq \mathcal{X}$ and w_k is disturbance with a known probability distribution (density) function, $p_k(w_k)$. We assume that the state x_k is directly observable with no uncertainty at time k, and the initial state x_0 is given. We define a *control policy* as a map, $\mu_k : \mathcal{X} \mapsto \mathcal{X}_k(x_k)$. A policy sequence is denoted by:

$$\boldsymbol{\mu} = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}.$$

Our objective is, given an initial state x_0, to find an optimal policy sequence $\boldsymbol{\mu}^\star$ over a finite control horizon $k = 0, \ldots, N$ that achieves the followings:

1) **Satisfaction of a joint chance constraint:** The probability that the state stays within a feasible region $\mathcal{X}_k \subseteq \mathcal{X}$ over the control horizon is at least $1 - \Delta$, where $\Delta \in [0, 1]$ is a user-specified *risk bound*.
2) **Minimization of a cost function:** We minimize the expected total cost over the control horizon, given a one-stage cost function $g_k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ and a terminal cost function $g_N : \mathcal{X} \mapsto \mathbb{R}$.

This constrained optimal control problem is formally stated as follows:

Problem 1: Joint Chance-constrained Optimal Control

$$\min_{\boldsymbol{\mu}} \mathbb{E}\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k)) \right\} \tag{1}$$

$$\text{s.t.} \quad \Pr\left\{ \bigcap_{k=1}^{N} x_k \in \mathcal{X}_k \,\middle|\, x_0 \right\} \geq 1 - \Delta. \tag{2}$$

This problem cannot be solved by the existing constrained DP/MDP approaches, such as Lagrangian-based methods [19], [21], because the left hand side of (2) has a form different from that of the objective function (1).

B. Reformulation though Risk Allocation

We address the challenge stated above by reformulating the joint chance constraint (2) into a constraint over an expectation of a summation of indicator functions, so that a Lagrangian-based approach can be applied. The indicator function $I_k(x_k)$ is defined as follows.

$$I_k(x_k) = \begin{cases} 1 & (x_k \notin \mathcal{X}_k) \\ 0 & (\text{Otherwise}) \end{cases} \tag{3}$$

In other words, $I_k(x_k)$ is one if x_k is *infeasible*. Using this indicator function, we formulate the following approximation of Problem 1:

Problem 2: Approximation of Problem 1

$$\min_{\boldsymbol{\mu}} \mathbb{E}\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k)) \right\} \tag{4}$$

$$\text{s.t.} \quad \mathbb{E}\left\{ \sum_{k=1}^{N} I_k(x_k) \,\middle|\, x_0 \right\} \geq \Delta. \tag{5}$$

The following theorem holds:

Theorem 1: Feasibility of Approximate Solution
A feasible solution to Problem 2 is also a feasible solution to Problem 1.

Proof: It suffices to show that (5) is a sufficient condition of (2). The probability of at each time step is represented by the indicator function $I_k(x_k)$:

$$\Pr\{x_k \notin \cup_k \|x_0\| = \mathbb{E}\{I_k(x_k) \|x_0\|.$$

Using the above equation, the left hand side of (2) is bounded as follows:

$$\begin{aligned}\Pr\left\{\bigvee_{k=1}^{N} x_k \notin \cup_k \|x_0\right\} &= 1 - \Pr\left\{\bigcap_{k=1}^{N} x_k \notin \cup_k \|x_0\right\} \\ &\sim 1 - \int_{k=1}^{N} \Pr\{x_k \notin \cup_k \|x_0\| \\ &= 1 - \int_{k=1}^{N} \mathbb{E}\{I_k(x_k) \|x_0\| \\ &= 1 - \mathbb{E}\left\{\int_{k=1}^{N} I_k(x_k) \|x_0\right\} \end{aligned} \tag{6}$$

Note that the inequality is derived from Boole's inequality: $\Pr[A\{ B] \geq \Pr[A]+\Pr[B]$. (6) implies that (5) is a sufficient condition of (2). ■

The same approximation is employed by the risk allocation approach [13], which is originally developed for chance-constrained MPC. It has been shown in the context of the risk allocation approach that, although Problem 2 is a conservative approximation of Problem 1, the conservatism due to this approximation is practically very small. More specifically, it has been shown [23] that, with an assumption that failures occur independently between time steps, the conservatism measured by the difference between the left hand side and the right hand side of the inequality in (6) is $\mathcal{U}(\Delta^2)$. In most practical cases, the risk bound Δ is set to a small value, such as 1%. If $\Delta = 1\%$, then the conservatism is about 0.01%. Furthermore, it is demonstrated by the previous work [13], [24] that the risk allocation approach results in substantially smaller suboptimality compared to other approximation approaches, such as [8], [9], [10], [25].

III. Method

We solve Problem 2 using a Lagrangian-based approach, which is similar to [19], [21], [26]. Our method is distinct from these existing approaches in that it allows users to explicitly specify a tolerance on the suboptimality of the dual optimization (Theorem 2-1). A suboptimality bound on the primal optimization is also obtained from the dual solution (Theorem 2-2). This capability is enabled by the newly designed stopping condition of the dual optimization, (13). It is particularly important to bound the solution error in our problem since the dual objective function is often non-differentiable, and it is difficult to obtain the exact solution.

A. Dual Optimization

Our approach is to solve the dual optimization problem of Problem 2. To this end, we first formulate the Lagrangian. For a dual variable $\lambda \sim 0$, let

$$L_k^\lambda(x_k, u_k) = \begin{cases} g_0(x_0, u_0) & (k = 0) \\ g_k(x_k, u_k) + \lambda I_k(x_k) & (k = 1 \ldots N - 1) \\ g_N(x_N) + \lambda I_N(x_N) & (k = N). \end{cases}$$

Then, the dual optimization problem is formulated as follows:

Problem 3: Dual of Problem 2

$$\max_{\lambda \geq 0} \min_{\boldsymbol{\mu}} \mathbb{E}\left\{\int_{k=0}^{N} L_k^\lambda(x_k, \mu_k(x_k))\right\} - \lambda\Delta.$$

We introduce the following simplified notation:

Problem 3': Dual of Problem 2 (Simplified notation)

$$\max_{\lambda \geq 0} \; q(\lambda),$$

where $q(\lambda)$ is the dual objective function, which can be evaluated by solving the following optimization problem with a given λ:

Problem 4: Evaluation of Dual Objective Function

$$J_0^\lambda(x_0) = \min_{\boldsymbol{\mu}} \mathbb{E}\left\{\int_{k=0}^{N} L_k^\lambda(x_k, \mu_k(x_k))\right\}.$$

Since Problem 4 is an unconstrained optimization, it can be solved efficiently by a standard DP method for a given λ. More specifically, the following backward recursion is conducted:

$$J_N^\lambda(x_N) := L_N^\lambda(x_N) \tag{7}$$

$$\begin{aligned} J_k^\lambda(x_k) := \min_{u_k \in U_k(x_k)} \mathbb{E}_{w_k} \{ & L_k^\lambda(x_k, u_k) \\ & + J_{k+1}(f(x_k, u_k, w_k))\}, \\ & k = 0, 1, \ldots N - 1. \end{aligned} \tag{8}$$

After the backward recursion, the dual objective function is obtained as:

$$q(\lambda) = J_0^\lambda(x_0) - \lambda\Delta.$$

Note that the dual objective function $q(\lambda)$ does not have to be obtained in a closed form, because the optimality conditions can be obtained without a closed-form expression of $q(\lambda)$, as explained in the next subsection.

B. Optimality Condition

The goal of this subsection is to obtain an optimality condition for Problem 3. Before proceeding, let us define three additional notations. First, let $\lambda^\star$ be the optimal dual solution for Problem 3. Second, we denote by $\boldsymbol{\mu}^\lambda$ the optimal solution for Problem 4 with a given λ, where

$$\boldsymbol{\mu}^\lambda = \{\mu_0^\lambda, \mu_1^\lambda, \ldots, \mu_N^\lambda\}.$$

Third, we define a *risk-to-go function*, $r_0^\lambda(x_0)$, which is equivalent to the left hand side of (5) given the optimal policy $\boldsymbol{\mu}^\lambda$:

$$r_0^\lambda(x_0) := \mathbb{E}\left[\sum_{k=1}^{N} I_k(x_k) \mid x_0, \boldsymbol{\mu}^\lambda\right]. \tag{9}$$

Intuitively, $r_0^\lambda(x_0)$ represents the conditional probability of failure[1] when the optimal policy $\boldsymbol{\mu}^\lambda$ is applied starting from the given initial state x_0. The risk-to-go function is computed by the following backward recursion:

$$r_N^\lambda(x_N) := I_N(x_N)$$
$$r_k^\lambda(x_k) := I_k(x_k) + \int_{w_k} r_{k+1}^\lambda(f(x_k, \mu_k^\lambda, w_k))p_k(w_k)dw_k$$

If the distribution is discrete, the integral is replaced with a summation.

We are now ready to formally discuss the optimality condition. It is known that the dual objective function $q(\lambda)$ is guaranteed to be concave [27]. Therefore, the maximum is achieved if:

$$0 \in \partial q(\lambda), \tag{10}$$

where $\partial q(\lambda)$ is the subgradient of $q(\lambda)$. If $0 \notin \partial q(\lambda)$ for all $\lambda > 0$, then $\lambda = 0$ is the optimal solution. Such a special case is handled separately in the proposed algorithm. It is also known that the subgradient contains:

$$r_0^\lambda(x_0) - \Delta \in \partial q(\lambda). \tag{11}$$

If q is differentiable at λ, then

$$\frac{dq}{d\lambda} = r_0^\lambda(x_0) - \Delta.$$

Therefore, $r_0^\lambda(x_0) - \Delta = 0$ is a sufficient condition for dual optimality. Hence, if q is differentiable at $\lambda^\star$, then Problem 3 can be optimally solved by a root-finding algorithm, such as the bisection method or Brent's method.

However, particularly when the control space $\mathcal{X}_k(x_k)$ is discrete, the maximum is often attained at an non-differentiable point of q, where $r_0^\lambda(x_0)$ is discontinuous. In such case, it is generally difficult to find the optimal solution by a root-finding algorithm. Instead, we run a root-finding algorithm until it finds an interval $[\underline{\lambda}, \overline{\lambda}]$ that satisfies:

$$r_0^{\underline{\lambda}}(x_0) - \Delta > 0, \quad r_0^{\overline{\lambda}}(x_0) - \Delta \geq 0 \tag{12}$$
$$(\underline{\lambda} - \overline{\lambda})\{r_0^{\overline{\lambda}}(x_0) - \Delta\} \geq \epsilon_d \tag{13}$$

for a given tolerance $\epsilon_d > 0$. Such an interval can be efficiently found by a root-finding algorithm since r_0^λ is monotonically non-increasing with λ. Standard root-finding algorithms, such as the bisection method and Brent's method, have exponential convergence rate [22]. Therefore, an interval that satisfies (12) and (13) can be found very efficiently. We empirically demonstrate the exponential convergence of the dual optimization in Section V-B (Figure 5). Note that the chance constraint (5), which is equivalent to $r_0^\lambda(x_0) - \Delta \geq 0$, is satisfied at $\overline{\lambda}$. Therefore, we use $\overline{\lambda}$ as an approximate solution to Problem 3. In the next subsection we show that the approximation error in the dual objective function is bounded by ϵ_d.

[1]Strictly speaking, the risk-to-go function $r_0^\lambda(x_0)$ does *not* represent the probability of failure, but an upper bound of it, because of the reformulation from Problem 1 to Problem 2.

C. Suboptimality Bounds

Our dual optimization approach may be suboptimal with regard to Problem 2 due to the following two factors:

1) Approximation error of a dual solution (i.e., $\overline{\lambda} - \lambda^\star$), as discussed in the previous subsection, and
2) Duality gap.

As for the first factor, the approximation error in the dual objective function is $q^\star - q(\overline{\lambda}) \sim 0$, where $q^\star$ is the optimal dual objective value. Regarding the second factor, even if $\lambda^\star = \overline{\lambda}$, the optimal dual objective value may not agree with the optimal primal objective value due to a duality gap. Therefore, the control policy obtained from the approximate solution to the dual optimization, $\boldsymbol{\mu}^{\overline{\lambda}}$, can be different from the optimal solution to Problem 2. We denote by h^λ the optimal objective value of the primal problem (Problem 2) given λ. Hence,

$$h^\lambda = q(\lambda) - \lambda(r_0^\lambda(x_0) - \Delta). \tag{14}$$

We also denote by $h^\star$ the optimal primal objective value to Problem 2. Then, since (12) guarantees that the primal solution for $\overline{\lambda}$ is feasible but not necessarily optimal, there is an error in the resulting primal objective value: $h^{\overline{\lambda}} - h^\star \sim 0$.

The following theorem provides bounds on these errors.

Theorem 2: Suboptimality Bounds

Let $q^\star$ and $h^\star$ be the optimal dual and primal objective values, respectively. Also, let $\overline{\lambda}$ be the approximate dual solution that satisfies (12) and (13). Then, the following holds:

1) Suboptimality of the dual objective value is bounded by:
$$q^\star - q(\overline{\lambda}) \geq \epsilon_d. \tag{15}$$
2) Suboptimality of the primal objective value is bounded by:
$$h^{\overline{\lambda}} - h^\star \geq \epsilon_p, \tag{16}$$
where
$$\epsilon_p := (\overline{\lambda} - \underline{\lambda})\{r_0^{\overline{\lambda}}(x_0) - \Delta\}.$$

Proof:

1) The dual objective function q is concave. Therefore,
$$q^\star \geq q(\overline{\lambda}) + d \times (\lambda^\star - \overline{\lambda}), \quad \forall d \in \partial q(\overline{\lambda}).$$
Since $q(\lambda)$ is concave and r_0^λ is monotonically non-increasing, it follows from (10), (11), and (12) that
$$\underline{\lambda} \geq \lambda^\star \geq \overline{\lambda}.$$
We use (11) and (13) to obtain:
$$\begin{aligned} q^\star &\geq q(\overline{\lambda}) + (\lambda^\star - \overline{\lambda})\{r_0^{\overline{\lambda}}(x_0) - \Delta\} \\ &\geq q(\overline{\lambda}) + (\underline{\lambda} - \overline{\lambda})\{r_0^{\overline{\lambda}}(x_0) - \Delta\} \geq q(\overline{\lambda}) + \epsilon_d. \end{aligned}$$

Therefore, (15) holds.

2) Since $q^\star$ and $h^\star$ are the optimal dual and primal objective values, $q(\overline{\lambda}) \geq q^\star \geq h^\star$. Therefore,

$$h^{\overline{\lambda}} \quad h^\star \geq h^{\overline{\lambda}} \quad q(\overline{\lambda}) = \quad \overline{\lambda}(r_0^{\overline{\lambda}}(x_0) \quad \Delta) = \epsilon_p.$$

The first equality follows from (14). ■

IV. Algorithm

We now present the algorithm that obtains an approximate solution to solve Problem 2. More specifically, the algorithm computes $\overline{\lambda}$ that satisfies (12) and (13), as well as the policy sequence $\boldsymbol{\mu}^{\overline{\lambda}}$. Recall that the feasibility of $\boldsymbol{\mu}^{\overline{\lambda}}$ is guaranteed by (12), and a bound on the approximation error is provided by Theorem 2. The proposed algorithm is described as below.

Algorithm 1 Joint Chance-Constrained Dynamic Programming

1: Solve Problem 4 with $\lambda = 0$
2: **if** $r_0^{\lambda}(x_0) \quad \Delta \geq 0$ **then**
3: **return** $\boldsymbol{\mu}^0$
4: **end if**
5: Solve Problem 5
6: **if** $\Delta_{\min} > \Delta$ **then**
7: **return** Infeasible
8: **end if**
9: $[\underline{\lambda} \ \overline{\lambda}] \rightarrow [0 \ \lambda^+]$
10: **while** $(\underline{\lambda} \quad \overline{\lambda})\}r_0^{\overline{\lambda}}(x_0) \quad \Delta| > \epsilon_d$ **do**
11: $\lambda \rightarrow$ Brent's method with $[\underline{\lambda} \ \overline{\lambda}]$
12: Solve Problem 4 with λ
13: **if** $r_0^{\lambda}(x_0) \quad \Delta = 0$ **then**
14: **return** $\boldsymbol{\mu}^{\lambda}$
15: **else if** $r_0^{\lambda}(x_0) \quad \Delta < 0$ **then**
16: $\overline{\lambda} \rightarrow \lambda$
17: **else**
18: $\underline{\lambda} \rightarrow \lambda$
19: **end if**
20: **end while**
21: **return** $\boldsymbol{\mu}^{\overline{\lambda}}$

Lines 1 - 4 are to separate the special case, $\lambda = 0$ (See the discussion in Section III-B). If the chance constraint is satisfied with $\lambda = 0$, then it is the optimal solution.

Lines 5 - 8 checks if a feasible solution to the primal optimization problem (Problem 2) exists by solving the following optimization problem:

Problem 5: Feasibility Check

$$\Delta_{\min} = \min_{\boldsymbol{\mu}} \mathbb{E} \left\{ \sum_{k=1}^{N} I_k(x_k) \ \| x_0 \right\}.$$

Note that the objective function of Problem 5 is the same as the constraint of Problem 2. Hence, $\Delta_{\min}$ means the minimum risk that can be achieved by any possible policy. Since Problem 5 is an unconstrained optimization problem, it can be efficiently solved by a standard DP method. If $\Delta_{\min}$ is larger than the specified risk bound Δ, then Problem 2 is infeasible. Otherwise, there exists a λ^+ such that $r_0^{\lambda^+}(x_0) \quad \Delta \geq 0$. In Line 9, $\overline{\lambda}$ is initialized with such a λ^+.

Lines 10 - 20 are the main loop of the algorithm. Line 11 computes one step of a root-finding algorithm, Brent's method, in order to obtain $\lambda \in (\underline{\lambda} \ \overline{\lambda})$. Then, in Line 12, the optimal policy $\boldsymbol{\mu}^{\lambda}$ is obtained by solving Problem 4 with this λ. Lines 13 - 17 updates $[\underline{\lambda} \ \overline{\lambda}]$ so that (12) is always satisfied. The algorithm terminates if (13) is satisfied, and returns the optimal policy with $\overline{\lambda}$.

V. Simulation results

We demonstrate the proposed algorithm on two types of problems: path planning and Mars EDL. The algorithm is implemented in MATLAB. Computation time is evaluated on a machine with an Intel Core 2 CPU clocked at 2.93 GHz and a 2 GB of memory.

A. Path Planning

In this example, we consider a two-dimensional rectangular state space, which is discretized into a 100x100 grid with an interval of 1.

The following dynamics are assumed:

$$x_{k+1} = x_k + u_k + w_k$$
$$u_k \ _2 \geq d_k, \quad w_k \leftarrow \mathcal{O}(0, \sigma^2 I),$$

where d_k and σ are constant parameters, $\mathcal{O}(0, \Sigma)$ is a zero-mean Gaussian distribution with the covariance matrix Σ, and I is the two-dimensional identity matrix. We set $d_k = 6$ and $\sigma = 1$ for Figure 2(a), and $d_k = 5$ and $\sigma = 1.67$ for Figure 2(b) and Table I. The control input and disturbance are also discretized with the same interval as the state variable.

The dynamic programming problem is formulated with 50 time steps ($N = 50$). We choose the locations of the start x_0 and the goal x_G randomly. The terminal cost is:

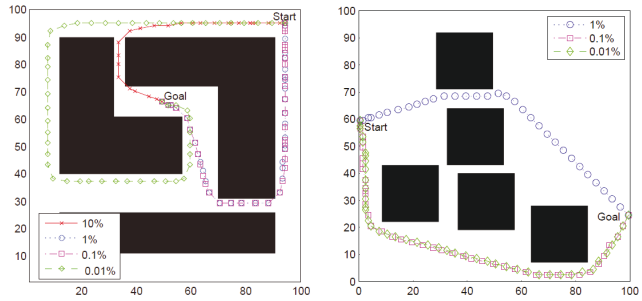
$$g_N(x_N) = \begin{cases} 0 & (\text{if } x_N = x_G) \\ 1 & (\text{Otherwise}), \end{cases}$$

while the stage cost is proportional to the path length of each step:

$$g_k(x_k, u_k) = \alpha \ u_k \ ,$$

where α is a constant. This constant must be set to a very small value in order to avoid a trivial solution that stays at the start at the all time steps. We use $\alpha = 10^{-5}$.

An illustrative example of the path planning problem is shown in Figure 2(a). The lines shown in the figure are the nominal paths with different risk bounds Δ, while the black blocks represent infeasible state regions. Here, a nominal path means a state sequence $x_0, \ldots, x_N$ that is obtained by applying the resulting control policy $\boldsymbol{\mu}^{\lambda}$ to the system without disturbances. When a 10% risk of failure is allowed, the nominal path goes through a narrow gap between the obstacles in order to minimize the path length. With 1% and 0.1% risk bounds, the nominal paths go through a wider gap in order to avoid excessive risk. When the risk bound is 0.01%, an even longer nominal path is chosen.



(a) (b)

Fig. 2. Application of the proposed chance-constrained dynamic programming algorithm to path planning problems with 50 time steps.

Next, we run the proposed algorithm in a state space with five randomly placed rectangular obstacles. Figure 2(b) shows an example of the state space as well as the resulting nominal paths. The simulation is run 100 times with three different risk bounds. The means and the standard deviations of the cost function values and the computation times are shown in Table I. The change in cost between different Δ is relatively small because the stage cost (i.e., path length) is significantly smaller than the terminal cost (i.e., penalty of failure to reach the goal at the final time step), due to the very small value of α.

TABLE I

THE AVERAGES AND THE STANDARD DEVIATIONS OF THE COSTS AND COMPUTATION TIMES FOR DIFFERENT RISK BOUNDS. FOR EACH CASES, 100 SIMULATIONS ARE CONDUCTED WITH RANDOM LOCATION OF OBSTACLES.

Risk bound	Cost	Length of nominal path	Computation time [sec]
$\Delta = 1\%$	0.88551 ± 0.0176	82.58 ± 25.10	17.2 ± 5.4
$\Delta = 0.1\%$	0.88555 ± 0.0175	86.74 ± 27.94	14.9 ± 4.6
$\Delta = 0.01\%$	0.88556 ± 0.0175	87.21 ± 28.11	12.2 ± 4.7

B. Mars EDL Scenario

We next demonstrate the proposed algorithm on the Mars EDL scenario shown in Figure 1. We employ the same dynamics model as [2], except that we assume stochastic disturbance at all time steps while [2] assumed set bounded disturbance at the PDG and HDA stages. At the kth stage, x_k represents the projected landing location without further control, as shown as the dashed lines in Figure 1. By applying a control at the kth stage, the lander can correct the projected landing location to u_k, which must be within an ellipsoid centered around x_k. At the end of the kth control stage, the projected landing location x_{k+1} deviates from u_k due to a disturbance w_k, which is assumed to have a Gaussian distribution. x_3 is the final landing location. This EDL model is described as follows:

$$x_{k+1} = u_k + w_k$$
$$(u_k \quad x_k)^T D_k (u_k \quad x_k) \geq d_k^2, \quad w_k \leftarrow \mathcal{O}(0, \Sigma_k),$$

where D_k and Σ_k are positive definite matrices, and d_k is a scalar constant. In this simulation, D_k is set to be the 2-D identity matrix, and d_k is set as follows:

$$d_0 = 3000 \text{ m}, \quad d_1 = 20 \text{ m}, \quad d_2 = 6 \text{ m}.$$

We assume that the covariance matrix Σ_k is a diagonal matrix with all diagonal elements being σ_k^2, where σ_k is a standard deviation. The 3-σ of each stage is:

$$3\sigma_0 = 500 \text{ m}, \quad 3\sigma_1 = 10 \text{ m}, \quad 3\sigma_2 = 2 \text{ m}.$$

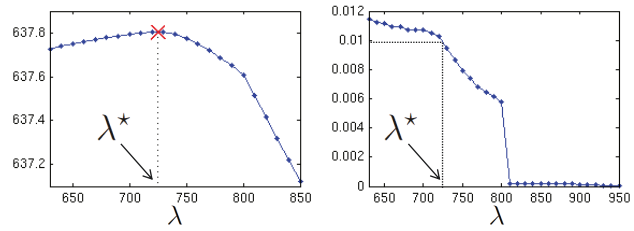
The state space $\cup$ is a 2 km-by-2 km square, which is discretized at a one meter resolution. As a result, the problem has four million states at each time step. The control and the disturbance are also discretized at the same resolution. The infeasible areas are specified using the data of HiRISE (High Resolution Imaging Science Experiment) camera on the Mars Reconnaissance Orbiter. We use the real landscape of a site named "East Margaritifer" on Mars.

Figure 4(a) shows the Lagrangian of the terminal stage, L_3^λ. The blue flat areas are infeasible areas for landing due to either steep slope or existence of obstacles, such as rocks. We only consider the terminal cost $g_N(x_N)$, which is equal to the minimum distance to travel in order to visit a specified number of science targets starting from the landing site. The method to obtain the minimum driving distance is described in detail in [2]. We place nine science targets, represented by squares in Figure 4(a) and labeled as A, B, ... I.

Figure 3(a) shows the dual objective function $q(\lambda)$ for a case with a 1% risk bound. The function is concave and achieves the maximum at $\overline{\lambda} = 725.2$. The probability of failure, $r_0^{\overline{\lambda}}(x_0)$, is 0.990% and is within the risk bound. The expected cost is $h^{\overline{\lambda}} = 637.81$ m. Using Theorem 2-2, the suboptimality bound on the expected cost is $\epsilon_p = 7.25 \pm 10^{-2}$ m. The optimal EDL target u_0 is shown in Figure 4(b) as well as a circle representing the three sigma of the disturbance w_0. The optimal EDL target is near the science target D.

With a smaller risk bound, $\Delta = 0.1\%$, the optimal EDL target moves to a location near the science target E, as shown in Figure 4(c). This is because, although the cost is relatively higher around the science target E than D, there are fewer obstacles in its proximity, and hence involves smaller risk of landing failure. As a result, the expected cost increases to $h^{\overline{\lambda}} = 644.82$ m, with a suboptimality bound of $\epsilon_p = 6.73 \pm 10^{-1}$ m. With an even smaller risk bound, $\Delta = 0.01\%$, the optimal EDL target location changes only slightly, as shown in Figure 4(c). The expected cost is $h^{\overline{\lambda}} = 645.54$ m, and the suboptimality bound is $\epsilon_p = 5.46 \pm 10^{-3}$ m.

An interesting thing to note is that, when the risk bound is $\Delta = 0.1\%$, the resulting probability of failure with the optimal policy is $r_0^{\overline{\lambda}}(x_0) = 0.0160\%$, which is significantly smaller than the given risk bound. This large gap between Δ and $r_0^{\overline{\lambda}}(x_0)$ is explained in Figure 3(b), which plots $r_0^{\lambda}(x_0)$ against λ. Note that the function is discontinuous at around $\lambda = 800$, which corresponds to a non-differentiable point of the dual objective function, shown in Figure 3(a). Since there is no λ that achieves $r_0^{\lambda}(x_0) = 0.1\%$, the algorithm chooses



(a) Dual objective $q(\lambda)$, $\Delta = 1\%$ (b) Risk-to-go function $r_0^\lambda(x_0)$

Fig. 3. (a) The dual objective function $p(\lambda)$ with $\Delta = 0.01$. Note that the function is concave, as predicted by the theory. The dual solution obtained by the proposed algorithm is $\overline{\lambda} = 725.16$. (b) The risk-to-go function $r_0^\lambda(x_0)$. Note the discontinuous feature of the plot. This is because, although the dual objective $p(\lambda)$ is continuous, it is not differentiable at countable number of points. The plot is generated by solving the primal problem with different values of λ with a uniform interval of 10.

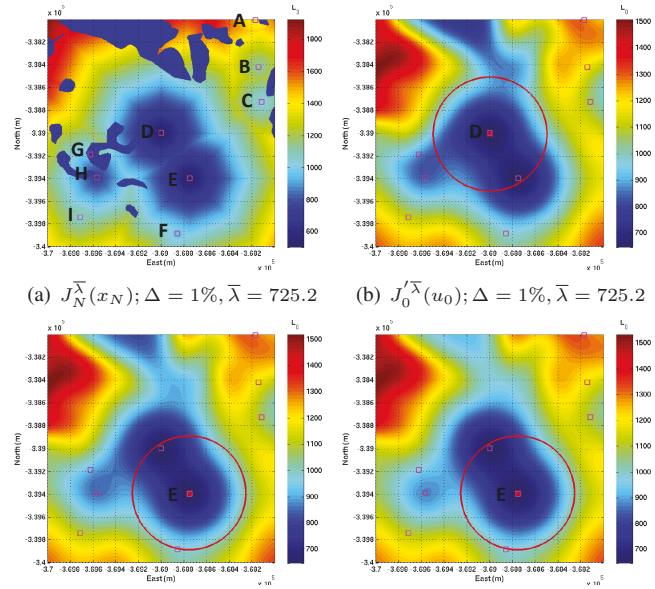
λ that is slightly right of the discontinuous point in order to satisfy the chance constraint. Such a discontinuous change in $r_0^\lambda(x_0)$ occurs due to a "jump" of the optimal EDL target from D to E, as shown in Figures 4(b) and 4(c). On the other hand, $r_0^\lambda(x_0)$ is nearly continuous when it crosses 0.01 and 0.0001. As a result, the probabilities of failure for $\Delta = 1\%$ and 0.01% are $r_0^{\overline{\lambda}}(x_0) = 0.990\%$ and 0.0094%, respectively, which are relatively close to the risk bounds.

The exponential convergence of the dual optimization (Lines 10-20 in Algorithm 1) is demonstrated in the semi-log plots in Figure 5. Note that a straight line in a semi-log plot represents an exponential relationship. In this simulation, we set the risk bound $\Delta = 0.1\%$ and the convergence tolerance $\epsilon_d = 10^{-3}$. Figure 5(a) plots the dual suboptimality bound, which corresponds to the left hand side of (13), against the number of dual iterations (i.e., the number of times Problem 4 is solved). The algorithm terminates when the dual suboptimality bound goes below ϵ_d. Figure 5(b) plots the width of the search interval of the zero-finding method, $\overline{\lambda} \quad \underline{\lambda}$. It is shown in the plots that both the suboptimality bound and the search interval decrease exponentially and converges with 23 iterations in this case.

In order to evaluate the computation time and the number of iterations, we run the algorithm 40 times with randomly located science targets. We set $\Delta = 0.1\%$ and $\epsilon_d = 10^{-3}$. The average and the standard deviation of the computation time are $188.1 \subseteq 76.1$ seconds, while those of the number of iterations are $16.1 \subseteq 5.8$.

C. Suboptimality bound

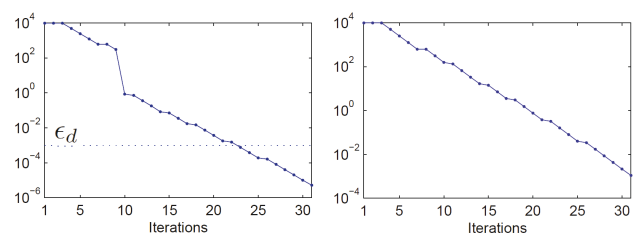
Finally, we empirically validate Theorem 2. We consider a variant of the Mars EDL scenario, where only one time step (EDL targeting) is considered. With this simplified problem setting, the exact optimal solution can be found by a brute-force approach (i.e., finding the best μ_0 among four million options). We compare the approximate primal objective value $h^{\overline{\lambda}}$, obtained from the proposed chance-constrained DP, with the optimal primal objective value $h^\star$, obtained from the brute-force approach. The simulation is run 100 times with randomized location of science targets and a risk bound $\Delta = 0.1\%$. Figure 6 plots the resulting suboptimality,



(a) $J_N^{\overline{\lambda}}(x_N)$; $\Delta = 1\%, \overline{\lambda} = 725.2$ (b) $J_0^{\prime\overline{\lambda}}(u_0)$; $\Delta = 1\%, \overline{\lambda} = 725.2$

(c) $J_N^{\prime\overline{\lambda}}(u_0)$; $\Delta = 0.1\%, \overline{\lambda} = 801.0$ (d) $J_N^{\prime\overline{\lambda}}(u_0)$; $\Delta = 0.01\%, \overline{\lambda} = 910.5$

Fig. 4. **(a)** The Lagrangian function $J_N^\lambda = L_N^\lambda(x_N)$ at the final stage with the dual solution $\overline{\lambda} = 725.2$. The blue flat areas are the infeasible regions (i.e., obstacles), penalized with a cost $\overline{\lambda}$. The Lagrangian values at the feasible locations represent the required distance to traverse after landing. Squares are science targets, to which the rover must drive after landing. **(b)-(d)** Expected cost at the initial stage as a function of u_0, $J_0^{\prime\overline{\lambda}}(u_0) := \mathbb{E}\{J_1^{\overline{\lambda}}(f(x_0, u_0, w_0))\}$, with $\Delta = 1\%$, 0.1%, and 0.01%, respectively. The red $\times$-mark is the optimal EDL target u_0, while the red circle represents 3σ of the disturbance in the first stage w_o. The dual solution $\overline{\lambda}$ is shown above each figure. The dimension of the map is 2000 x 2000 meters, which is discretized at a 1-meter resolution.



(a) $(\underline{\lambda} - \overline{\lambda})\{r_0^{\overline{\lambda}} - \Delta\}$ (b) $\overline{\lambda} - \underline{\lambda}$

Fig. 5. Exponential convergence of the dual optimization. (a) The dual error bound, $(\underline{\lambda} - \overline{\lambda})\{r_0^{\overline{\lambda}}(x_0) - \Delta\}$, which gives an upper bound on the error in the dual objective value (Theorem 2). The stopping condition (13) requires that the dual error is below ϵ_d. (b) The width of the search interval $[\underline{\lambda} \ \overline{\lambda}]$, in which the dual optimal solution $\lambda^\star$ is guaranteed to exist. Note that the plots are in a semi-log scale.

$h^{\overline{\lambda}} \quad h^\star$, against the suboptimality bound given by Theorem 2, $\epsilon_p = \quad \overline{\lambda}\}r_0^{\overline{\lambda}}(x_0) \quad \Delta|$. In all the 100 runs, the error is less than the error bound. Furthermore, in 24 runs, the suboptimality is exactly zero, meaning that the solution of the proposed algorithm is the exact optimal solution.

VI. Conclusion

This paper presented a novel chance-constrained dynamic programming algorithm, which outputs a control policy that minimizes the expected cost while guaranteeing that the probability of constraint violation is within a user-specified risk bound. Through the careful reformulation of the problem

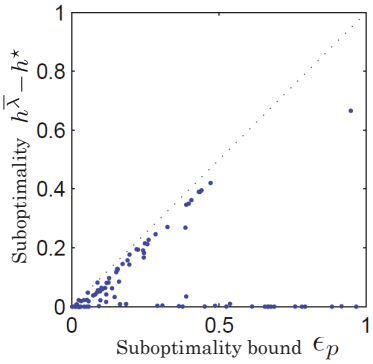


Fig. 6. Suboptimality of the primal objective value, as well as the primal suboptimality bounds, of 100 solutions of the simplified Mars EDL scenario with randomly placed scientific targets. The vertical axis of the plot represents the observed suboptimality, $h^{\overline{\lambda}} - h^{\star}$, while the horizontal axis represents the primal suboptimality bound, ϵ_p. Theorem 2-2 is empirically validated by the fact that all samples are below the 45° line, shown in the dotted line.

using the dual optimization, the original problem is converted to a combination of a standard DP and a root-finding problem that is solved iteratively. Although the obtained solution is suboptimal in nature, the suboptimality bound is explicitly characterized and is shown to be very small when the acceptable probability of failure is set to be small. Applications to Mars EDL analysis and path planning are shown in simulation, together with the numerical verification of the presented theories.

Acknowledgment

Marco Pavone provided insightful feedback and contributed to implementing the algorithm. The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] P. C. Knocke, G. G. Wawrzyniak, B. M. Kennedy, P. N. Desai, T. J. Parker, M. P. Golombek, T. C. Duxbury, and D. M. Kass, "Mars exploration rovers landing dispersion analysis," in *Proceedings of AIAA/AAS Astrodynamics Specialist Conferencec and Exhibit*, 2004.

[2] Y. Kuwata, M. Pavone, and J. B. Balaram, "A risk-constrained multi-stage decision making approach to the architectural analysis of mars missions," in *Proceedings of the IEEE Conference on Decision and Control*, 2012.

[3] Y. Kuwata and J. B. Balaram, "Combined EDL-mobility planning for planetary missions," in *Proceedings of Infotech@Aerospace*, 2011.

[4] B. Acikmese and S. Ploen, "Convex programming approach to powered descent guidance for mars landing," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.

[5] A. Johnson, A. Huertas, R. Werner, and J. Montgomery, "Analysis of On-Board Hazard Detection and Avoidance for Safe Lunar Landing," in *IEEE Aerospace Conference*, 2008.

[6] D. P. Bertsekas, *Dynamic Programming and Optimal Control Volume I (Third Edition)*. Athena Scientific, 2005.

[7] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1743–1752, 1999. [Online]. Available: http://dx.doi.org/10.1002/aic.690450811

[8] 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.

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

[10] 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.

[11] P. Couchman, M. Cannon, and B. Kouvaritakis, "Stochastic MPC with inequality stability constraints," *Automatica*, vol. 42, no. 12, pp. 2169 – 2174, 2006.

[12] F. Oldewurtel, C. N. Jones, and M. Morari, "A tractable approximation of chance constrained stochastic mpc based on affine disturbance feedback," in *Proceedings of Conference on Decision and Control*, 2008.

[13] 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*, 2008.

[14] M. Ono, "Joint chance-constrained model predictive control with probabilistic resolvability," in *Proceedings of American Control Conference*, 2012.

[15] F. Oldewurtel, A. Ulbig, A. Parisio, G. Andersson, and M. Morari, "Reducing peak electricity demand in building climate control using real-time pricing and model predictive control," in *Proceedings of Conference on Decision and Control*, 2010.

[16] A. E. Sjödin, D. F. Gayme, and U. Topcu, "Risk-mitigated optimal power flow with high wind penetration," in *Proceedings of American Control Conference*, 2012.

[17] A. J. Askew, "Optimum reservoir operating policies and the imposition of a reliability constraint," *Water Resources Research*, 1974.

[18] ——, "Chance-constraind dynamic programming and the optimization of water resource systems," *Water Resources Research*, 1974.

[19] L. A. Rossman, "Reliability-constrained dynamic programing and randomized release rules in reservoir management," *Water Resources Research*, 1977.

[20] E. Altman, *Constrained Markov decision processes*, ser. Stochastic modeling. Chapman & Hall/CRC, 1999. [Online]. Available: http://books.google.com/books?id=3X9S1NM2iOgC

[21] F. J. Beutler and K. W. Ross, "Optimal policies for controlled markov chains with a constraint," *Journal of Mathematical Analysis and Applications*, 1985.

[22] K. E. Atkinson, *An Introduction to Numerical Analysis, Second Edition*. John Wiley & Sons, 1989.

[23] M. Ono, "Robust, goal-directed plan execution with bounded risk," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.

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

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

[26] J. L. Williams, I. John W. Fisher, and A. S. Willsky, "Approximate dynamic programming for communication-constrained sensor network management," *IEEE Transactions on Signal Processing*, vol. 55, 2007.

[27] D. P. Bertsekas, *Convex Optimization Theory*. Athena Scientific, 2009.