Noname manuscript No.
(will be inserted by the editor)

Chance-Constrained Dynamic Programming with Application to Risk-Aware Robotic Space Exploration

Masahiro Ono · Marco Pavone · Yoshiaki Kuwata · J. (Bob) Balaram

Received: date / Accepted: date

Abstract Existing approaches to constrained dynamic programming are limited to formulations where the constraints share the same additive structure of the objective function (that is, they can be represented as an expectation of the summation of one-stage costs). As such, these formulations cannot handle joint probabilistic (chance) constraints, whose structure is not additive. To bridge this gap, this paper presents a novel algorithmic approach for joint chance-constrained dynamic programming problems, where the probability of failure to satisfy given state constraints is explicitly bounded. Our approach is to (conservatively) reformulate a joint chance constraint as a constraint on the expectation of a summation of indicator random variables, which can be incorporated into the cost function by considering a dual formulation of the optimization problem. As a result, the primal variables can be optimized by 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 algorithm effectiveness on three optimal control problems, namely a path planning problem, a Mars entry, descent and landing problem, and a Lunar landing problem.

Masahiro Ono
Jet Propulsion Laboratory, California Institute of Technology
E-mail: ono@jpl.nasa.gov

Marco Pavone
Stanford University, Department of Aeronautics and Astronautics
E-mail: pavone@stanford.edu

Yoshiaki Kuwata
Space Exploration Technologies Corporation (SpaceX)
E-mail: kuwata@gmail.com

J. (Bob) Balaram
Jet Propulsion Laboratory, California Institute of Technology
E-mail: j.balaram@jpl.nasa.gov

Fig. 1 Autonomous landing of the Curiosity rover on Mars using the Sky Crane maneuver. (Image credit: NASA/JPL-Caltech)

All Mars simulations are conducted using real terrain data of Mars, with four million discrete states at each time step. The numerical experiments are used to validate our theoretical and heuristic arguments that the proposed algorithm is both (i) computationally efficient, i.e., capable of handling real-world problems, and (ii) near-optimal, i.e., its degree of conservatism is very low.

Keywords Dynamic programming · Constrained Stochastic optimal control · Chance-constrained optimization · Markov decision processes · Path planning

1 Introduction

1.1 Problem description and motivation

Autonomy is poised to play an increasingly important role in robotic space exploration. Indeed, autonomy is already playing an important role in current Mars missions. For example, the entry, descent, and landing (EDL) phase of the 2012 Mars Science Laboratory (MSL) mission, which involved a complex Skycrane maneuver [37] (see Figure 1),

was performed completely autonomously, as the 14-minute speed-of-light delay between Earth and Mars exceeded the 7-minute duration of the EDL phase. Autonomous path planning and hazard avoidance enabled Mars rovers to go beyond the sight of ground operators [9]. Also, an automated scientific data collection system called the Autonomous Exploration for Gathering Increased Science (AEGIS) [16] is being used onboard the Opportunity rover.

On the other hand, risk management in most space missions is largely a manual process. In the Mars Exploration Rovers (MER) mission, for example, ground operators decided whether or not to perform a trajectory correction maneuver before atmospheric entry by checking if the probability of safe landing was above a pre-specified threshold, which was set to 91% for Spirit and 96% for Opportunity [21]. As an additional example, when driving Mars rovers through rocky terrain, paths are pre-planned by ground operators with careful consideration of the uncertainty in rover position.

In order to continue advancing the frontier of robotic space exploration, there is a clear need for an on-board risk management capability. Indeed, in the Keck Institute Space Studies (KISS) Workshop on Resilient Space Systems in 2012, experts from NASA centers and academia concluded that risk-aware on-board decision making will significantly benefit the future missions recommended by NASA's Planetary Science Decadal Survey [28], including Comet Surface Sample Return, Saturn Probe, and Trojan Tour and Rendezvous [26]. The workshop also identified chance-constrained planning as a core component of the proposed Resilient Spacecraft Executive, as demonstrated later on in [25].

Specifically, in many applications, risk requirements are formulated as bounds on the probability of success, as exemplified by the 91% and 96% bounds previously described for MER EDL. Among the various risk-aware decision making approaches such as [12,43,14,17], the chance-constrained optimization framework, originally proposed in [13], naturally fits this formulation, as one can explicitly impose a bound on the probability of success in a stochastic problem setting.

Accordingly, this paper focuses on chance-constrained dynamic programming (CCDP) for three reasons. First, like standard DP, the solution to CCDP is a closed-loop control policy, which explicitly maps states into control inputs. Given a stochastic state transition model, an optimal control policy can be computed off-line, stored in a look-up table, and then executed in real-time even on computationally-limited vehicles such as spacecraft. Second, CCDP can perform sequential decision making over multiple time steps. For example, future Mars EDL maneuvers will aim at rejecting uncertainty by active feedback control in three different stages: entry-phase targeting, powered-descent guidance (PDG) [1,**?**,**?**,**?**], and hazard detection and avoidance (HDA) [20]. Third, CCDP can also be used for trade analysis. For example, in a mission to reach a specific target on a planetary surface, increased rover mobility would relax the requirement on landing accuracy. In fact, in our previous work, we designed the Combined EDL and Mobility Analysis Tool (CEMAT) to perform such a trade based on a DP formulation [23,22].

1.2 Literature review

CCDP was initially studied in the 1970s [4,3], in the context of water management. These studies, however were field-specific and lacked a theoretical justification. A problem similar to CCDP, which entails dynamic programming under "reliability constraints," was studied in [38]. In this formulation, the objective was to limit the expected *number* of failures and the approach was to employ a Lagrangian method to transform the constrained problem into an unconstrained counterpart. More generally, a number of diverse approaches have been proposed to address constrained Markov Decision Processes (MDP), including linear programming and Lagrangian methods. An authoritative overview is provided in [2]. A major assumption in the existing literature on constrained MDP is that the constraints must share the same additive structure of the objective function (that is, they can be represented as an expectation of the summation of one-stage costs). Unfortunately, this assumption precludes the application of existing tools to the problem of chance-constrained MDP (since the structure of chance constraints is not additive), as also pointed out in [38]. One possibility to circumvent this issue is to consider penalty-based MDPs, whereby one achieves risk aversion by imposing an arbitrary cost penalty on failure states. Unfortunately, stakeholders usually desire *explicit* constraints on the probability of failure, as is the case for most space missions.

Over the last decade, due to major breakthroughs in embedded optimization and a need to explicitly bound failure probabilities, chance-constrained optimization has been intensively studied within the Model Predictive Control (MPC) community [40,18,24,10,15,30,35,32], under the name of Chance-Constrained MPC (CCMPC). Successful applications include building climate control [31,46] and electrical power grids [34,41,42,45]. A chance-constrained extension to standard optimal control methods such as LQR and LQG has also been studied in [19,39]. From this perspective, the objective of this paper is to integrate into dynamic programming (DP) the rich insights obtained from the MPC community, and develop a general CCDP framework that can be applied to a broad range of stochastic optimal control problems[1].

[1] The proposed algorithm is distinct from MPC in that, as a regular DP, it computes a control policy, instead of a sequence of control inputs, backward in time.

1.3 Contributions

The contributions of this paper are threefold. First, we propose an algorithm for CCDP, whereby a joint chance constraint is (conservatively) transformed into an expectation over a summation of indicator random variables, which enjoys the same additive structure of the cost function and can be incorporated into the cost function by considering a dual formulation of the optimization problem. The algorithm evaluates the dual objective function by minimizing the Lagrangian via standard dynamic programming (i.e., Bellman's recursion), given a fixed dual variable. The dual variable is then optimized via a root-finding algorithm, namely the Brent's method [5], which has an exponential convergence rate and a complexity that does not change with primal problem's size. Simulations show that the algorithm typically converges within 10 to 30 iterations.

Second, we derive bounds on the suboptimality of both the primal and dual objective values, which provide rigorous stopping criteria for the root finding method. The technical hurdle in our analysis is that the dual objective function is, in general, non-differentiable.

Finally, we demonstrate our algorithm on a path planning problem, a Mars entry, descent and landing problem, and a Lunar landing problem. Mars simulations are conducted using real terrain data of Mars, with four million discrete states at each time step. These numerical experiments support our theoretical and heuristic arguments that the proposed algorithm is both (i) computationally efficient, i.e., capable of handling real-world problems, (ii) near-optimal, i.e., its degree of conservatism is very low.

The rest of the paper is organized as follows. In Section 2 we present known concepts from optimization, on which we will rely extensively in this paper. In Section 3 we formulate the problem we wish to solve, i.e., CCDP, together with a reformulation that allows us to apply dual optimization techniques. In Section 4 we present a dual formulation of CCDP, exact and approximate optimality conditions, related suboptimality bounds, and, finally, the proposed algorithm for CCDP. In Section 5 we present numerical experiments. Finally, we draw our conclusions in Section 6.

2 Background

Given a convex function $f(x) : \mathbb{R}^n \to \mathbb{R}$, a vector $d \in \mathbb{R}^n$ is said to be a *subgradient* of f at a point $x \in \mathbb{R}^n$ if

$$f(z) \geq f(x) + (z - x)'d, \qquad \text{for all } z \in \mathbb{R}^n. \tag{1}$$

If instead f is a concave function, a vector d is said to be a subgradient of f at x if $-d$ is a subgradient of the convex function $-f$ at x [6, Appendix B]. The set of all subgradients of a convex (or concave) function f at $x \in \mathbb{R}^n$ is referred to as the subdifferential of f at x, and is denoted by $\partial f(x)$. Intuitively, subdifferentiability is a generalization of differentiability to non-differentiable functions. In fact, one can show that f is differentiable at x with gradient $\nabla f(x)$ if and only if it has $\nabla f(x)$ as its unique subgradient at x [6, Appendix B].

A fundamental result in convex optimization is that a point x minimizes a convex function f over a convex set $X \subset \mathbb{R}^n$ if and only if there exists a subgradient $d \in \partial f(x)$ such that ([6, Appendix B])

$$d'(z - x) \geq 0, \qquad \text{for all } z \in X. \tag{2}$$

Equation (2) generalizes the optimality condition for the case where f is differentiable, that is $\nabla f(x)'(z - x) \geq 0$ for all $z \in X$. In the special case where $X = \mathbb{R}^n$, one obtains a basic necessary and sufficient condition for unconstrained optimality of x:

$$0 \in \partial f(x).$$

We next discuss basic properties concerning dual formulations of optimization problems. Consider the optimization problem

$$\begin{aligned} \min \quad & f(x) && (3)\\ \text{subject to} \quad & g_j(x) \leq 0, \quad j = 1, \ldots, r, && (4) \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g_j : \mathbb{R}^n \to \mathbb{R}$ are given functions. We refer to this problem as the primal problem and we denote its value with f^*. The dual of the above problem is given by

$$\begin{aligned} \max \quad & q(\lambda) && (5)\\ \text{subject to} \quad & \lambda_j \geq 0, \quad j = 1, \ldots, r, && (6) \end{aligned}$$

where

$$q(\lambda) := \inf_{x \in \mathbb{R}^n} \left\{ f(x) + \sum_{j=1}^{r} \lambda_j \, g_j(x) \right\}.$$

The function $q(\lambda)$ is referred to as the dual function. The function $L(x, \lambda) := f(x) + \sum_{i=1}^{r} \lambda_j \, g_j(x)$ is referred to as the Lagrangian. We denote the the value of the dual problem as q^*. Let $\lambda := (\lambda_1, \ldots, \lambda_r)$.

It turns out that the dual problem is always a concave problem (concave cost, convex constraint set) even if the primal is *not* convex [6, Chapter 6]. An important result connecting primal and dual problems is the weak duality theorem, according to which $q^* \leq f^*$ [6, Chapter 6]. If $q^* = f^*$ we say that there is no duality gap, while if $q^* < f^*$ we say that there is a duality gap.

For a given $\lambda \in \mathbb{R}^r$, let x_λ be a value minimizing the Lagrangian, i.e.,

$$x_\lambda := \arg\min_{x \in \mathbb{R}^n} \; L(x, \lambda).$$

A useful fact is that $g(x_\lambda)$ is a subgradient of the dual function q at λ, that is $g(x_\lambda) \in \partial q(\lambda)$ [6, Chapter 6]. This result will be exploited in Section 4.

3 Formulation of Joint Chance-Constrained DP

3.1 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 dynamical model:

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

where u_k is a control input constrained to belong to a possibly state-dependant control set $\mathcal{U}_k(x_k)$, $\mathcal{U}$ is the control space, w_k is a disturbance with a known probability distribution (density) function $p_k(w_k)$, and N is the number of decision stages. 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* μ_k as a map from states to controls, i.e., $\mu_k : \mathcal{X} \mapsto \mathcal{U}_k(x_k)$ for $k = 0, \dots, N$. A policy sequence is denoted by:

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

Given an initial state x_0, the objective is to find an optimal policy sequence $\boldsymbol{\mu}^\star$ that achieves the following:

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 N is at least $1 - \Delta$, where $\Delta \in [0,\ 1]$ is a user-specified *risk bound*.
2. **Minimization of a cost function:** Given a one-stage cost function $g_k : \mathcal{X} \times \mathcal{U} \mapsto \mathbb{R}$ and a terminal cost function $g_N : \mathcal{X} \mapsto \mathbb{R}$, the expected total cost over the control horizon is minimized.

Once $\boldsymbol{\mu}^\star$ is obtained, the optimal control input for a given state can be immediately obtained on-line just by a table look-up. Therefore the focus of this paper is given to an off-line computation of the optimal policy.

The problem we are interested in can then be formulated as follows:

Problem 1: Joint Chance-Constrained Optimal Control

$$\min_{\boldsymbol{\mu}} \quad \mathbb{E}\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k)) \right\} \tag{7}$$

$$\text{subject to} \quad \Pr\left\{ \bigwedge_{k=1}^{N} x_k \in \mathcal{X}_k \;\middle|\; x_0 \right\} \geq 1 - \Delta. \tag{8}$$

Existing constrained DP/MDP methods, such as those presented in [38,8], only consider the case where the constraint function is in the same form as the objective function (i.e., terminal cost plus summation over one-stage costs). Hence, they are not directly applicable to Problem 1 as the left hand side of (8) has a form different from that of the objective function (7); in other words, it does not have an additive structure.

3.2 Reformulation via Boole's Approximation

Our technical approach is to reformulate the joint chance constraint (8) into a constraint over an expectation of a summation of indicator random variables, so that a Lagrangian-based approach can be applied. An indicator random variable $I_k(x_k)$ is defined as follows:

$$I_k(x_k) := \begin{cases} 1, & \text{if } x_k \notin \mathcal{X}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

In other words, $I_k(x_k)$ is equal to one if x_k is *infeasible*. Using indicator random variables and Boole's inequality, Problem 1 can be approximated as follows:

Problem 2: Approximation of Problem 1

$$\min_{\boldsymbol{\mu}} \quad \mathbb{E}\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k)) \right\} \tag{10}$$

$$\text{subject to} \quad \mathbb{E}\left\{ \sum_{k=1}^{N} I_k(x_k) \;\middle|\; x_0 \right\} \leq \Delta. \tag{11}$$

The following theorem holds:

Theorem 1 (Conservatism of Problem 2) *Problem 2 is a conservative approximation of Problem 1. In other words, a feasible solution to Problem 2 is guaranteed to be a feasible solution to Problem 1.*

Proof We prove this theorem by showing that (11) is a sufficient condition for (8). Indeed, the probability of the event $\{x_k \notin \mathcal{X}_k\}$ is equal to the expectation of the random variable $I_k(x_k)$, that is

$$\Pr\{x_k \notin \mathcal{X}_k \mid x_0\} = \mathbb{E}\{I_k(x_k) \mid x_0\}.$$

Using the above equation, the left hand side of (8) can be lower bounded as

$$\begin{aligned} \Pr\left\{ \bigwedge_{k=1}^{N} x_k \in \mathcal{X}_k \mid x_0 \right\} &= 1 - \Pr\left\{ \bigvee_{k=1}^{N} x_k \notin \mathcal{X}_k \mid x_0 \right\} \\ &\geq 1 - \sum_{k=1}^{N} \Pr\{x_k \notin \mathcal{X}_k \mid x_0\} \\ &= 1 - \sum_{k=1}^{N} \mathbb{E}\{I_k(x_k) \mid x_0\} \\ &= 1 - \mathbb{E}\left\{ \sum_{k=1}^{N} I_k(x_k) \mid x_0 \right\}. \end{aligned} \tag{12}$$

In the second step we used Boole's inequality, i.e., $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$. Equation (12) implies that (11) is a sufficient condition for (8), which proves the theorem.

Note that in Problem 2 the constraint has an *additive* structure, hence Problem 2 is much easier to solve than Problem 1. Accordingly, our approach is to solve Problem 2 as a conservative approximation to Problem 1. Specifically, Problem 2 falls within the class of constrained Markov Decision Processes (CMDP), discussed extensively in [2]. In this paper we exploit the fact that Problem 2 has a single constraint, and design a tailored, exponentially-converging solution approach that relies on root-finding methods. The Boole's approximation used in the derivation of Problem 2 is the same approximation used in [35] within the context of chance-constrained MPC. It has been shown that the conservatism introduced by this approximation is, in practice, very small. More specifically, it has been shown that, under the 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 (12) is of the order of $O(\Delta^2)$ [33]. In most practical cases, the risk bound Δ is set to a small value, which implies (at least under the assumption of independent failures) that the degree of conservatism is moderate. This aspect will be verified numerically in Section 5. Furthermore, previous works [11, 35] showed that the Boole's approximation yields substantially less conservatism compared to other approaches, such as those in [10, 18, 24, 29].

4 Method

In this section we develop a computationally-efficient algorithm to solve Problem 2. Our approach, which relies on dual optimization, is inspired by a number of earlier works on constrained MDP [8, 38, 44]. The key difference is that in this paper we derive conditions that allow users to *explicitly* specify a tolerance for the suboptimality of the dual optimization problem. This is important as the dual objective function is often non-differentiable, and hence it is difficult to obtain an exact solution.

Specifically, in this section we first formulate the dual of Problem 2 (Section 4.1). Then, we proceed to a discussion of exact and approximate optimality conditions (Section 4.2). Finally, exploiting the suboptimality bounds derived in Section 4.3, we present an algorithm that provides a conservative, approximate solution to Problem 2 (Section 4.4). This solution, in turn, represents a conservative, approximate solution for Problem 1 (CCDP).

4.1 Dual of Problem 2

To formulate the dual of Problem 2, we define a step-wise Lagrangian as

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

where $\lambda \geq 0$ is a dual variable. With this definition, the dual of Problem 2 can be written as

Problem 3: Dual of Problem 2

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

The objective function in Problem 3 can be written as $q(\lambda) := J_0^\lambda(x_0) - \lambda\Delta$, where

$$J_0^\lambda(x_0) := \min_{\boldsymbol{\mu}} \mathbb{E} \left\{ \sum_{k=0}^{N} L_k^\lambda(x_k, \mu_k(x_k)) \right\}. \tag{13}$$

Thus, Problem 3 can be compactly written as $\max_{\lambda \geq 0} \; q(\lambda)$. Note that, for a given λ, $J_0^\lambda(x_0)$ can be efficiently solved via standard dynamic programming. Specifically, $J_0^\lambda(x_0)$ can be computed by performing the following backward recursion:

$$J_N^\lambda(x_N) = L_N^\lambda(x_N), \tag{14}$$

$$J_k^\lambda(x_k) = \min_{u_k \in U_k(x_k)} \mathop{\mathbb{E}}_{w_k} \left\{ L_k^\lambda(x_k, u_k) + J_{k+1}\left(f(x_k, u_k, w_k)\right) \right\}, \tag{15}$$

for $k = 0, 1, \dots N-1$. In the next section we discuss optimality conditions for the dual objective function $q(\lambda)$, which will be exploited to derive an iterative algorithm for the solution of Problem 2.

4.2 Optimality and Approximate Optimality Conditions for the Dual Problem

The goal of this subsection is to obtain approximate optimality conditions for Problem 3, which will then be leveraged in our iterative solution algorithm. Let us start with three additional definitions. First, let $\lambda^\star$ be an optimal dual solution for Problem 3. Second, for a given λ, let $\boldsymbol{\mu}^\lambda$ be an optimal solution for the optimization problem in equation (13), where

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

Finally, we define a *risk-to-go function*, $r_0^\lambda(x_0)$, as the left hand side of equation (11) given the optimal policy $\boldsymbol{\mu}^\lambda$, that is

$$r_0^\lambda(x_0) := \mathbb{E} \left\{ \sum_{k=1}^{N} I_k(x_k) \mid x_0, \boldsymbol{\mu}^\lambda \right\}. \tag{16}$$

In the rest of the paper, we simply denote the risk-to-go function by r_0^λ, as the initial state is assumed to be given and equal to x_0. Intuitively, r_0^λ represents the conditional probability of failure[2] when the optimal policy $\boldsymbol{\mu}^\lambda$ is applied starting from the given initial state x_0. The risk-to-go function can be computed via the following backward recursion:

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

for $k = 0, 1, \dots N-1$. If the distribution is discrete, the integral should be replaced with a summation. It is straightforward to verify that r_0^λ is *monotonically not increasing* with respect to λ. Note that, once a dual solution λ is given, the value of the primal problem (i.e., Problem 2) corresponding to a policy $\boldsymbol{\mu}^\lambda$, denoted by h^λ, can be readily computed as

$$h^\lambda = q(\lambda) - \lambda(r_0^\lambda - \Delta). \tag{17}$$

We are now in a position to discuss optimality conditions for Problem 3. According to the results in Section 2, the dual problem is concave, in particular the dual objective function $q(\lambda)$ is concave (the constraint set, that is $\lambda \geq 0$, is clearly convex). Note that $q(\lambda)$ is possibly non-differentiable. According to equation (2), a dual variable λ^* maximizes $q(\lambda)$ over $\lambda \geq 0$ if and only if there exists a subgradient $d \in \partial q(\lambda^*)$ such that

$$d'(\lambda - \lambda^*) \leq 0, \qquad \text{for all } \lambda \geq 0. \tag{18}$$

(Recall that $q(\lambda)$ is concave, so the sign in equation (2) is reversed.) From the discussion at the end of Section 2, one has

$$r_0^\lambda - \Delta \in \partial q(\lambda). \tag{19}$$

To gain intuition about our proposed algorithm, assume that $\lambda^* > 0$ (see Figure 2). In this case, equations (18) and (19) imply that a necessary and sufficient condition for dual optimality is

$$0 \in \partial q(\lambda). \tag{20}$$

This motivates our approach whereby we use a root-finding algorithm, such as the bisection method or Brent's method, to iteratively compute an interval, denoted by $[\lambda^L,\ \lambda^U]$, such that

$$0 < r_0^{\lambda^L} - \Delta, \quad r_0^{\lambda^U} - \Delta \leq 0, \text{ and} \tag{21}$$
$$(\lambda^L - \lambda^U)(r_0^{\lambda^U} - \Delta) \leq \epsilon_d. \tag{22}$$

Given equations (19) and (20) and the fact that r_0^λ is monotonically not increasing, equation (21) ensures that $\lambda^* \in [\lambda^L,\ \lambda^U]$ (see Figure 2). Equation (22) is a complementary slackness condition ensuring that the approximation error in the dual objective function is bounded by an arbitrarily small design constant ϵ_d – this fact will be rigorously proven in Theorem 2. Figure 3 represents graphically the role of the complementary slackness condition. Note that the optimal policy corresponding to λ^L (slightly) violates the chance constraint, while the optimal policy corresponding to λ^U respects it. Hence, once the bisection algorithms has found an interval $[\lambda^L,\ \lambda^U]$ that satisfies the complementary slackness condition, we use λ^U as a conservative, approximate solution to Problem 3.

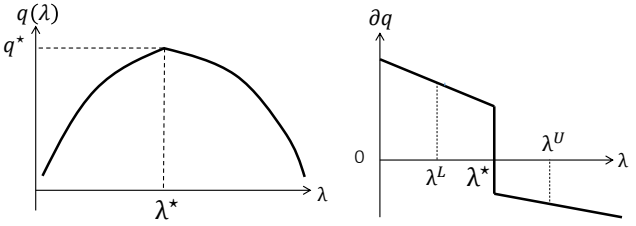


Fig. 2 Dual objective function (left) and its subgradient (right). The dual objective function is always concave. The subgradient contains zero at the optimum, $\lambda^\star$.

The special case $\lambda^* = 0$ is treated separately by the solution algorithm (see Lines 1–4 in Algorithm 1). We note that the case $\lambda^* = 0$ corresponds to the trivial case where the constraint is not active (i.e., the solution to the unconstrained problem is also a solution to the constrained problem).

Many standard root-finding algorithms, including the bisection method and Brent's method, have demonstrated exponential convergence rates [5]. This translates into an exponential convergence rate for our iterative algorithm for dual optimization. Numerical experiments confirming this statement will be provided in Section 5.2 (Figure 8). We highlight the advantage of this approach over subgradient methods [7], which represent a general solution approach for dual optimization problems. The convergence of subgradient methods is known to be very slow - it requires $O(1/\epsilon^2)$ iterations to find an ϵ-suboptimal solution. In contrast, the bisection method requires only $O(\log_2(1/\epsilon))$ iterations, and the Brent's method is at least as fast as the bisection method. A root-finding algorithm can be readily used in our case as we reduced the joint chance-constrained optimization problem to a special case of a CMDP with a single constraint (Problem 2).

Next, we prove in Section 4.3 that indeed equation (22) ensures that the approximation error for the dual objective function is bounded by ϵ_d, and we also provide a suboptimality bound for the primal objective. Then, in Section 4.4 we provide an algorithm that computes (with an exponential convergence rate) an interval $[\lambda^L,\ \lambda^U]$ fulfilling conditions

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

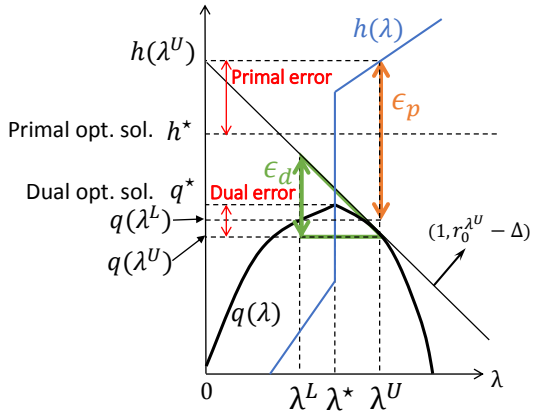


Fig. 3 Graphical interpretation of Theorem 2. The numbers ϵ_p and ϵ_d bound the primal and dual errors, respectively.

(21) and (22), and hence provides an approximate solution to Problem 2 with provable suboptimality bounds.

4.3 Suboptimality Bounds

The approximate primal solution, h^{λ^U}, is in general different from the optimal solution of Problem 2, denoted by $h^\star$. Since we pose Problem 2 as a minimization, $h^{\lambda^U} \geq h^\star$, as illustrated in Figure 3. The suboptimality in the approximate primal solution is due to the following two factors:

1. approximation error of the dual solution (i.e., $\lambda^U - \lambda^\star$), as discussed in the previous subsection, and
2. duality gap.

As for the first factor, the approximation error in the dual objective function is $q^\star - q(\lambda^U) \geq 0$, where $q^\star$ is the optimal dual objective value. Regarding the second factor, a duality gap exists in general unless Problem 2 is a convex optimization problem. The following theorem provides bounds on the primal and dual optimization errors.

Theorem 2 (Suboptimality Bounds) *Let λ^U be an approximate dual solution that satisfies equations* (21) *and* (22). *Then, the following holds:*

1. *the suboptimality of the dual objective value is bounded according to*

$$q^\star - q(\lambda^U) \leq \epsilon_d. \tag{23}$$

2. *the suboptimality of the primal objective value, corresponding to the feasible solution $\boldsymbol{\mu}_{\lambda^U}$, is bounded according to*

$$h^{\lambda^U} - h^\star \leq \epsilon_p, \tag{24}$$

where

$$\epsilon_p := \min\left(-\lambda^U(r_0^{\lambda^U} - \Delta), h^{\lambda^U} - h^{\lambda^L} - \lambda^L(r_0^{\lambda^L} - \Delta)\right).$$

Proof We start by proving the first claim. Since the dual objective function $q(\lambda)$ is concave, one has, for all subgradients $d \in \partial q(\lambda^U)$,

$$q^\star \leq q(\lambda^U) + (\lambda^\star - \lambda^U)\, d,$$

see equation (1). Since r_0^λ is monotonically non-increasing, equation (21) implies

$$\lambda^L \leq \lambda^\star \leq \lambda^U.$$

By using (19) and equation (22) one readily obtains

$$\begin{aligned} q^\star &\leq q(\lambda^U) + (\lambda^\star - \lambda^U)(r_0^{\lambda^U} - \Delta) \\ &\leq q(\lambda^U) + (\lambda^L - \lambda^U)(r_0^{\lambda^U} - \Delta) \leq q(\lambda^U) + \epsilon_d, \end{aligned}$$

which proves the first claim, that is equation (23).

We prove, now, the second claim. Since $q^\star$ and $h^\star$ are the optimal dual and primal objective values, by the weak duality theorem (see Section 2), one has

$$\max\left(q(\lambda^U), q(\lambda^L)\right) \leq q^\star \leq h^\star.$$

Therefore,

$$\begin{aligned} h^{\lambda^U} - h^\star &\leq \min\left(h^{\lambda^U} - q(\lambda^U), h^{\lambda^U} - q(\lambda^L)\right) \\ &= \min\Big(-\lambda^U(r_0^{\lambda^U} - \Delta), \\ &\qquad h^{\lambda^U} - h^{\lambda^L} - \lambda^L(r_0^{\lambda^L} - \Delta)\Big). \end{aligned}$$

where the equality follows from equation (17).

See Figure 3 for a graphical interpretation of Theorem 2. Note that $r_0^{\lambda^U} - \Delta$ is the slope of a tangent line to $q(\lambda)$ at λ^U.

4.4 An Approximation Algorithm for CCDP

We next present an algorithm to compute an approximate solution to Problem 2. Specifically, the algorithm computes a value λ^U that satisfies conditions (21) and (22), as well as the policy sequence $\boldsymbol{\mu}^{\lambda^U}$. Recall that the feasibility of $\boldsymbol{\mu}^{\lambda^U}$ is guaranteed by equation (21), and a bound on the approximation error is provided in Theorem 2.

The algorithm is given in pseudo-code in Algorithm 1. The algorithm starts by dealing with two special cases. First, Lines 1 – 4 consider the special case $\lambda^\star = 0$ (see the discussion in Section 4.2). Second, Lines 5 – 8 are aimed at identifying the case where no feasible solution to the primal optimization problem (Problem 2) exists. For this purpose, the algorithm solves (via standard dynamic programming) the optimization problem:

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

Note that the objective function in the above optimization problem is the same as that in the constraint of Problem 2. Hence, $\Delta_{\min}$ represents the minimum risk-to-go that can be achieved by any possible policy. If $\Delta_{\min}$ is larger than the specified risk bound Δ, then Problem 2 is infeasible.

If a given problem does not fall into the above two special cases, then there exists a λ^+ such that $r_0^{\lambda^+}(x_0) - \Delta \leq 0$. In Line 9, λ^U 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, namely Brent's method, in order to obtain $\lambda \in [\lambda^L,\ \lambda^U]$. Then, in Line 12, the optimal policy $\boldsymbol{\mu}^\lambda$ is obtained by computing $J_0^\lambda(x_0)$ with this λ. Lines 13 – 17 update $[\lambda^L,\ \lambda^U]$ so that (21) is always satisfied. The algorithm terminates whenever inequality (22) is satisfied, and returns the policy corresponding to λ^U.

By construction the algorithm is guaranteed to compute a dual variable λ^U that satisfies conditions (21) and (22), and hence a policy $\boldsymbol{\mu}^{\lambda^U}$ that is feasible for Problem 2 and with a rigorous suboptimality bound as given in Theorem 2.

The algorithm can be intuitively understood by regarding λ as a penalty of constraint violation. Then Line 12 of the algorithm can be viewed as solving an optimal control problem with a soft constraint. The level of penalty is iteratively changed by the Brent's method until the probability of constraint violation is right below the chance constraint.

Algorithm 1 Chance-Constrained Dynamic Programming

Input: Error tolerance $\epsilon_d > 0$
Output: Policy $\boldsymbol{\mu}^{\lambda^U}$ that is feasible for Problem 2 (if a feasible solution exists) and with suboptimality bounds given in Theorem 2.
1: Compute $J_0^0(x_0)$ {Special case for $\lambda^\star = 0$}
2: **if** $r_0^0(x_0) - \Delta \leq 0$ **then**
3: **return** $\boldsymbol{\mu}^0$
4: **end if**
5: Solve

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

{Feasibility check}
6: **if** $\Delta_{\min} > \Delta$ **then**
7: **return** Infeasible
8: **end if**
9: $[\lambda^L\ \lambda^U] \leftarrow [0,\ \lambda^+]$
10: **while** $(\lambda^L - \lambda^U)\{r_0^{\lambda^U}(x_0) - \Delta\} > \epsilon_d$ **do**
11: $\lambda \leftarrow$ Brent's method with $[\lambda^L,\ \lambda^U]$
12: Compute $J_0^\lambda(x_0)$ and obtain $\boldsymbol{\mu}^\lambda$
13: **if** $r_0^\lambda(x_0) - \Delta = 0$ **then**
14: **return** $\boldsymbol{\mu}^\lambda$
15: **else if** $r_0^\lambda(x_0) - \Delta < 0$ **then**
16: $\lambda^U \leftarrow \lambda$
17: **else**
18: $\lambda^L \leftarrow \lambda$
19: **end if**
20: **end while**
21: **return** $\boldsymbol{\mu}^{\lambda^U}$

5 Simulation results

We demonstrate the proposed algorithm on three problems: path planning, Mars EDL, and Lunar landing. 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 2 GB of memory.

5.1 Path Planning

In this example, we consider a two-dimensional rectangular state space discretized into a 100x100 grid, where the edge length of each cell corresponds to a unit length. The following dynamics are assumed:

$$x_{k+1} = x_k + u_k + w_k$$
$$\|u_k\|_2 \leq d_k, \quad w_k \sim \mathcal{N}(0, \sigma^2 I),$$

where d_k and σ are constant parameters, $\mathcal{N}(0, \Sigma)$ is a zero-mean Gaussian distribution with covariance matrix Σ, and I is the two-dimensional identity matrix. We set $d_k = 6$ and $\sigma = 1$ for Figure 4(a), and $d_k = 5$ and $\sigma = 1.67$ for Figure 4(b) and Table 1. The control input and disturbance are also discretized using the same interval as the state variable.

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

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

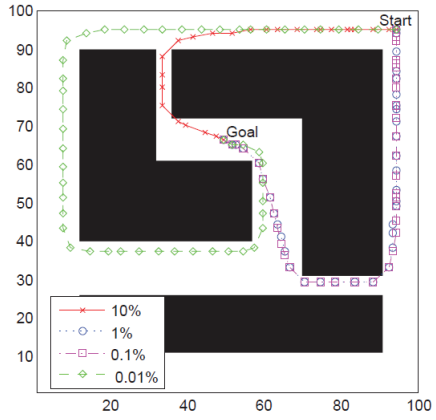
while the stage cost is proportional to the path length of each step, that is

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

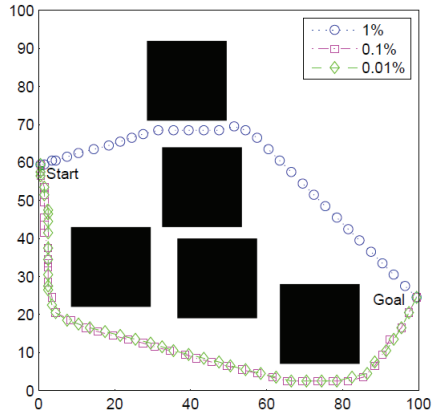
where $\alpha > 0$ is a constant. This constant must be set to a very small value in order to avoid a trivial solution that stays at the start state at all time steps. Here we use $\alpha = 10^{-5}$.

An illustrative example of the path planning problem is shown in Figure 4(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 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.

Next, we run the proposed algorithm in a state space with five randomly placed rectangular obstacles. Figure 4(b) shows an example of the state space as well as the resulting



(a)



(b)

Fig. 4 Application of the proposed chance-constrained dynamic programming algorithm to path planning problems with 50 time steps.

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 1. 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. Notice that the computation time is positively correlated to risk bound. Very roughly speaking, a problem with Δ that is close to zero is easy to solve because by setting λ arbitrary large, you can bring $r_0^{\lambda^U}$ close to zero. More precisely, a smaller risk bound results in a greater dual variable λ, and the risk to go $r_0^{\lambda^U}$ becomes less sensitive for a greater λ. (This can be intuitively understand from the fact that $\lim_{\lambda\to\infty} r_0^{\lambda^U} = 0$.) Note that the stopping condition is given as Line 10 of Algorithm 1,or equivalently, (22). For a small risk bound, the first few iterations of the Brent's method tend to bring $r_0^{\lambda^U} - \Delta$ sufficiently close to zero that the tolerance on $(\lambda^L - \lambda^U)$ greater, requiring smaller number of iterations of the Brent's method.

Table 1 Averages and standard deviations of the costs and computation times for different risk bounds. For each case, 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

Finally, in order to evaluate the conservatism of the proposed algorithm, we obtained the approximate solution, λ^U, for the map shown in Figure 4(b) with $\Delta = 0.1$, 0.01, and 0.001. The resulting risk-to-go ($r_0^{\lambda^U}$) as well as the actual probability of failure (P_{fail}) evaluated by Monte-Carlo simulations with 10,000 samples are shown in Table 2. As expected, for all cases, $\Delta > r_0^{\lambda^U} > P_{fail}$. The difference between Δ and $r_0^{\lambda^U}$ is due to the nonconvexity of the problem as well as the tolerance of the zero-finding method. The difference between $r_0^{\lambda^U}$ and P_{fail} is due to the conservative approximation using Boole's inequality in (12). The results in Table 2 are consistent with our claim that the conservatism introduced by the proposed algorithm is moderate. More importantly, since we only employed conservative approximations, the resulting solution always respects the given chance constraint.

Table 2 Empirical evaluation of the conservatism of the proposed algorithm. Optimal control policy is obtained for the map shown in Figure 4(b) with $\Delta = 0.1$, 0.01, and 0.001. The risk-to-go with the approximate optimal solution, λ^U, is shown in the second column. The actual probability of failure shown in the third column is evaluated by Monte-Carlo simulations with 10,000 samples.

Risk bound (Δ)	Risk-to-go ($r_0^{\lambda^U}$)	Prob. of failure
10%	8.70%	8.05%
1%	0.95%	0.81%
0.1%	0.09%	0.06%

5.2 Mars EDL Scenario

We next demonstrate the proposed algorithm on the Mars EDL scenario shown in Figure 5. As we discussed in the introduction, 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) [1], and hazard detection and avoidance (HDA) [20], as shown in Figure 5. 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.

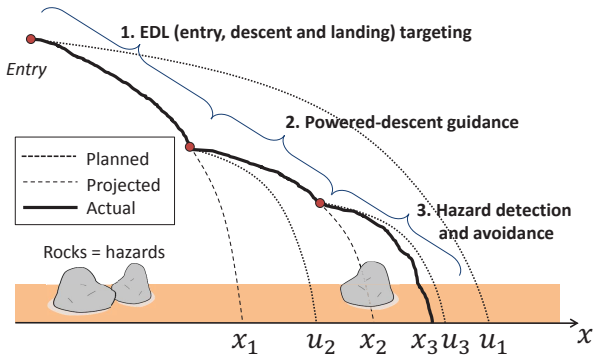


Fig. 5 A future Mars entry, descent, and landing scenario.

We employ the same dynamics model as [23], except that we assume stochastic disturbances at all stages while [23] assumed set bounded disturbances 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 5. 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. State x_3 is the final landing location. This EDL model is described as follows:

$$x_{k+1} = u_k + w_k,$$
$$(u_k - x_k)^T D_k (u_k - x_k) \leq d_k^2, \quad w_k \sim \mathcal{N}(0, \Sigma_k),$$

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 equal to σ_k^2, where σ_k is the 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 $\mathcal{X}$ 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 from the 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.

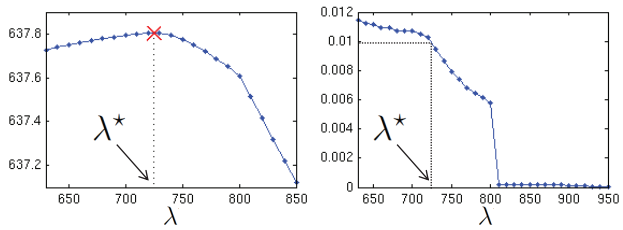
Figure 7(a) shows the Lagrangian of the terminal stage, L_3^λ. The blue flat areas are infeasible areas for landing due to either steep slope or existence of obstacles, such as rocks. We only consider the terminal cost $g_N(x_N)$, which is equal to the minimum driving distance in order to visit a specified number of science targets starting from the landing site. The method to obtain the minimum driving distance is described in detail in [23]. We place nine science targets, represented by squares in Figure 7(a) and labeled as A, B, ..., I.

Figure 6(a) shows the dual objective function $q(\lambda)$ for a case with a 1% risk bound. The function is concave and achieves its 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, the suboptimality bound on the expected cost is $\epsilon_p = 7.25 \times 10^{-2}$ m. The optimal EDL target u_0 is shown in Figure 7(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 science target E, as shown in Figure 7(c). This is because, although the cost around science target E is higher than around target D, there are fewer obstacles in its proximity, and hence target E involves a smaller risk of landing failure. As a result, the expected cost increases to $h^{\overline{\lambda}} = 644.82$ m, with a suboptimality bound of $\epsilon_p = 6.73 \times 10^{-1}$ m. With an even smaller risk bound, $\Delta = 0.01\%$, the optimal EDL target location changes only slightly, as shown in Figure 7(c). The expected cost is $h^{\overline{\lambda}} = 645.54$ m, and the suboptimality bound is $\epsilon_p = 5.46 \times 10^{-3}$ m.

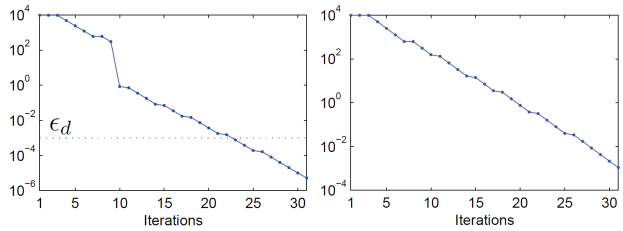
An interesting aspect to note is that, when the risk bound is $\Delta = 0.1\%$, the resulting probability of failure with the optimal policy is $r_0^{\overline{\lambda}}(x_0) = 0.0160\%$, which is significantly smaller than the given risk bound. Such a large gap between Δ and $r_0^{\overline{\lambda}}(x_0)$ is explained by Figure 6(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 6(a). Since there is no λ that achieves $r_0^{\lambda}(x_0) = 0.1\%$, the algorithm chooses a dual variable 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 7(b) and 7(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 Algorithm 1 is demonstrated in the semi-log plots in Figure 8. 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 8(a) plots the dual suboptimality bound, which corresponds to the left hand side of (22), against the number of dual iterations (i.e., the number of times $J_0^{\lambda}(x_0)$ is computed). The algorithm terminates when the dual suboptimality bound goes below



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

Fig. 6 (a) The dual objective function $q(\lambda)$ 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 $q(\lambda)$ is continuous, it is not differentiable at a countable number of points. The plot is generated by solving the primal problem with different values of λ with a uniform interval of 10.



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

Fig. 8 Exponential convergence of Algorithm 1. (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 (22) 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.

ϵ_d. Figure 8(b) plots the width of the search interval of the root-finding method, that is $\overline{\lambda} - \underline{\lambda}$. It is shown in the plots that both the suboptimality bound and the search interval decrease exponentially and converge within 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 ± 76.1 seconds, while the statistics of the number of iterations are 16.1 ± 5.8.

We also note that almost no preprocessing is required for our algorithm because it can directly take occupancy map as an input. However, generating an occupancy map for Mars applications is a labor intensive process, which typically requires manual identification of hazards on the images from Mars orbiters.

5.3 Lunar Landing Scenario

Optimal planning of Lunar landing is an interesting problem since the availability of low-energy transfer trajectories adds another dimension to the decision space. Low-energy transfer trajectories exploit the four-body dynamics between Earth, Moon, Sun, and the spacecraft [36]. As opposed to direct transfer trajectories, such as those used by the Apollo program, low-energy transfer trajectories require significantly smaller ΔV (i.e., less amount of propellant) but can involve significantly longer transfer times. Such trajectories were employed by multiple missions, including JAXA's Hiten in 1990 and NASA's GRAIL in 2011. Figure 9 shows the trajectories taken by the two GRAIL spacecraft, which took 112 and 113 days, respectively, to complete the transfer.

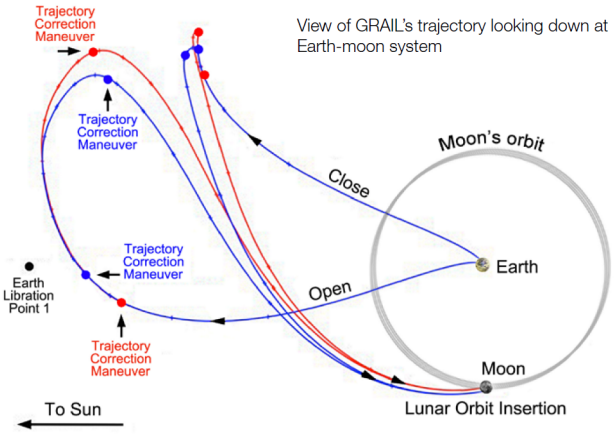


Fig. 9 Low-energy transfer trajectory used by GRAIL [27].

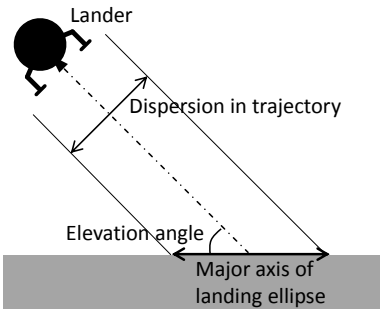
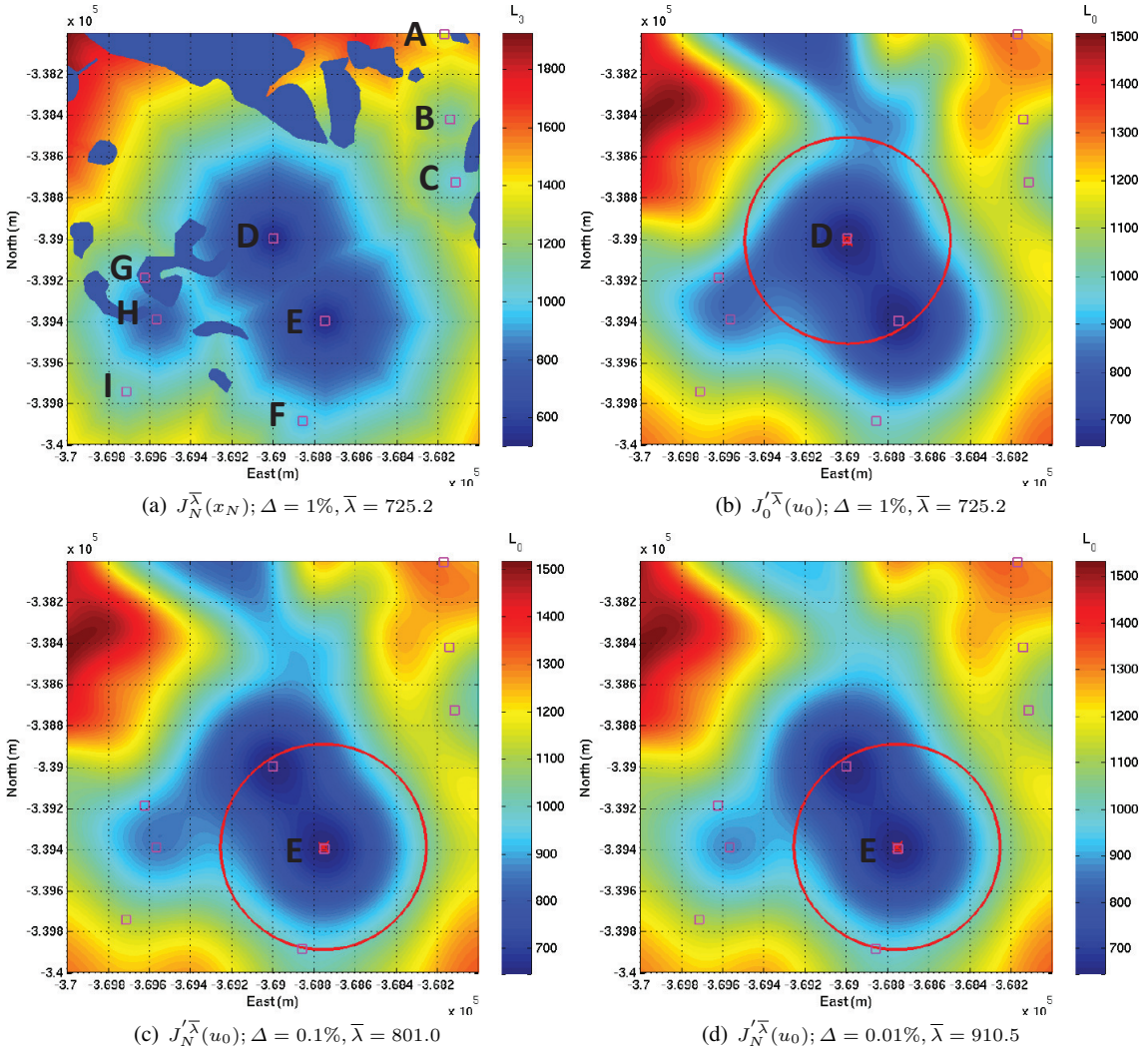


Fig. 10 At a given level of trajectory dispersion of a lander, the major axis of the landing ellipse becomes smaller by arriving at a greater elevation angle.

An additional benefit of low-energy transfer trajectories is that some of them allow a lander to arrive on the surface with a large elevation angle, resulting in smaller landing ellipses, as illustrated in Figure 10. In general, by allowing a longer transfer time, options with a greater elevation angle become available. The orientation of the ellipse is determined by the azimuth angle.

The resulting optimal Lunar landing problem is formulated as a CCDP with an additional decision variable with respect to the Mars EDL problem, namely, trajectory selec-



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

(b) $J_0'^{\overline{\lambda}}(u_0)$; $\Delta = 1\%, \overline{\lambda} = 725.2$

(c) $J_N'^{\overline{\lambda}}(u_0)$; $\Delta = 0.1\%, \overline{\lambda} = 801.0$

(d) $J_N'^{\overline{\lambda}}(u_0)$; $\Delta = 0.01\%, \overline{\lambda} = 910.5$

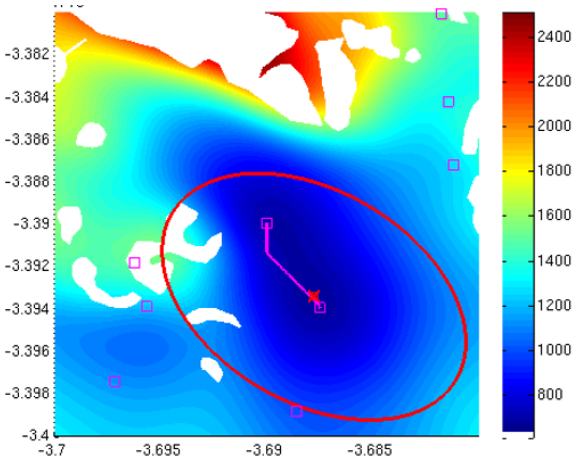
Fig. 7 **(a)** The Lagrangian function $J_N^{\lambda} = L_N^{\lambda}(x_N)$ at the final stage with the dual solution $\overline{\lambda} = 725.2$. The blue flat areas are the infeasible regions (i.e., obstacles), penalized with a cost $\overline{\lambda}$. The Lagrangian values at the feasible locations 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, $J_0'^{\overline{\lambda}}(u_0) := \mathbb{E}\{J_1^{\overline{\lambda}}(f(x_0, u_0, w_0))\}$, 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.

tion. In order to solve the problem, we first generate a table that contains a finite number of trajectory options. Each option specifies elevation and azimuth angles at arrival, transfer time, and required ΔV. In addition to the chance constraint, we also impose a constraint on the transfer time by removing the options for which the transfer times are greater than an upper bound. We use the same terrain as Section 5.2. The latitude and longitude of the landing site is assumed to be 0° and 310°, respectively. The risk bound is set to be $\Delta = 10\%$. With this setup, we use CCDP to obtain the optimal control policy with various settings of the upper bound on the transfer time.

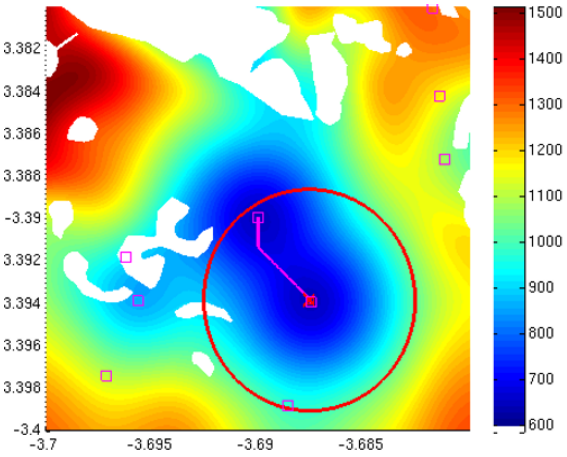
Figure 11 shows some sample results. With only 10 day transfer time allowed, the best elevation angle is 40°, resulting in the highly elliptical landing ellipse as shown in the figure. The expected surface driving distance to visit the two pre-specified targets is 676.5 m. On the other hand, when 100 day transfer time is allowed, a trajectory option with 73° elevation angle becomes available. Even though the landing target is not significantly different from the previous case, the smaller landing ellipse results in a reduced expected surface driving distance.

5.4 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



(a) $t_T \leq 10$ days, $\theta_E = 40°$, $\mathbb{E}\{d\} = 676.5$ m



(b) $t_T \leq 100$ days, $\theta_E = 73°$, $\mathbb{E}\{d\} = 648.5$ m

Fig. 11 Lunar landing problem results. t_T: transfer time to the Moon, θ_E: the elevation angle at arrival, d: driving distance to visit two of the given science targets.

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 dynamic programming algorithm, 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 12 plots the observed suboptimality, i.e., $h^{\overline{\lambda}} - h^\star$, against the suboptimality bound given by Theorem 2, $\epsilon_p = -\overline{\lambda}\,(r_0^{\overline{\lambda}}(x_0) - \Delta)$. 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.

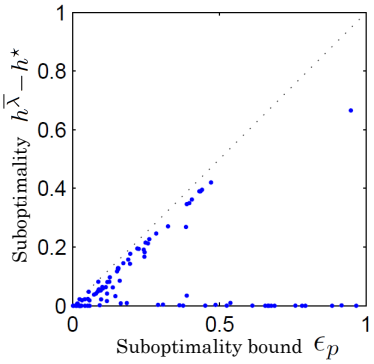


Fig. 12 Suboptimality of the primal objective value, as well as the primal suboptimality bound, of 100 solutions for the simplified Mars EDL scenario with randomly placed scientific targets. The vertical axis of the plot represents the observed suboptimality, that is $h^{\overline{\lambda}} - h^\star$, while the horizontal axis represents the primal suboptimality bound, ϵ_p. Theorem 2 is empirically validated by the fact that all samples are below the $45°$ line, shown with a the dotted line.

6 Conclusion

This paper presented a novel chance-constrained dynamic programming algorithm, which outputs a control policy that minimizes an expected cost while guaranteeing that the probability of constraint violation is within a user-specified risk bound. Through a careful reformulation of the problem using Boole's inequality and dual optimization, the original problem is converted into a combination of standard dynamic programming and root-finding problems, which are solved iteratively. Although the obtained solution is suboptimal, such suboptimality is practically quite moderate. Applications to path planning, Mars EDL, and Lunar landing are shown in simulation, together with the numerical verification of our theoretical results. Future work towards on-board deployment includes validation of the algorithm with more complex dynamics and probability distributions, as well as development of a compact representation of the control policy (e.g., through table look-ups) for systems with limited on-board resources. Another interesting direction would be to parameterize the policies to reduce memory requirement as well as to be able to handle continuous/infinite state space.

Acknowledgment

The research described in this paper was conducted at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Marco Pavone was supported in part by NASA under the Space Technology Research Grants Program, Grant NNX12AQ43G.

References

1. Acikmese, B., Ploen, S.: Convex programming approach to powered descent guidance for mars landing. AIAA Journal of Guidance, Control, and Dynamics **30**(5), 1353–1366 (2007)
2. Altman, E.: Constrained Markov decision processes. Stochastic modeling. Chapman & Hall/CRC (1999)
3. Askew, A.J.: Chance-constraind dynamic programming and the optimization of water resource systems. Water Resources Research (1974)
4. Askew, A.J.: Optimum reservoir operating policies and the imposition of a reliability constraint. Water Resources Research (1974)
5. Atkinson, K.E.: An Introduction to Numerical Analysis, Second Edition. John Wiley & Sons (1989)
6. Bertsekas, D.P.: Nonlinear programming (1999)
7. Bertsekas, D.P.: Nonlinear Programming, Second Edition. Athena Scientific (1999)
8. Beutler, F.J., Ross, K.W.: Optimal policies for controlled markov chains with a constraint. Journal of Mathematical Analysis and Applications (1985)
9. Biesiadecki, J.J., Maimone, M.: The mars exploration rover surface mobility flight software: Driving ambition. In: 2006 IEEE Aerospace Conference Proceedings (2006)
10. Blackmore, L., Li, H., Williams, B.C.: A probabilistic approach to optimal robust path planning with obstacles. In: Proceedings of American Control Conference (2006)
11. Blackmore, L., Ono, M.: Convex chance constrained predictive control without sampling. In: Proceedings of the AIAA Guidance, Navigation and Control Conference (2009)
12. Blackmore, L., Ono, M., Bektassov, A., Williams, B.C.: A probabilistic particle-control approximation of chance-constrained stochastic predictive control. IEEE Transactions on Robotics **26**(3), 502–517 (2010)
13. Charnes, A., Cooper, W.W.: Chance-constrained programming. Management Science **6**, 73–79 (1959)
14. Chow, Y.L., Pavone, M.: A framework for time-consistent, risk-averse model predictive control: Theory and algorithms. In: American Control Conference, pp. 4204–4211 (2014)
15. Couchman, P., Cannon, M., Kouvaritakis, B.: Stochastic MPC with inequality stability constraints. Automatica **42**(12), 2169 – 2174 (2006)
16. Estlin, T.A., Bornstein, B.J., Gaines, D.M., Anderson, R.C., Thompson, D.R., Burl, M., Castano, R., Judd, M.: Aegis: Automated science targeting for the mer opportunity rover. ACM Transactions on Intelligent Systems and Technology **3**(3) (2012)
17. Fleming, W., McEneaney, W.: Risk-sensitive control on an infinite time horizon. SIAM Journal on Control and Optimization **33**(6), 1881–1915 (1995)
18. van Hessem, D.H.: Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation. Ph.D. thesis, Delft University of Technology (2004)
19. Hokayem, P., Chatterjee, D., Lygeros, J.: Chance-constrained LQG with bounded contr ol policies. In: IEEE Conference on Decision and Control, pp. 2471–2476. Florence, Italy (2013)
20. Johnson, A., Huertas, A., Werner, R., Montgomery, J.: Analysis of On-Board Hazard Detection and Avoidance for Safe Lunar Landing. In: IEEE Aerospace Conference (2008)
21. Knocke, P.C., Wawrzyniak, G.G., Kennedy, B.M., Desai, P.N., Parker, T.J., Golombek, M.P., Duxbury, T.C., Kass, D.M.: Mars exploration rovers landing dispersion analysis. In: Proceedings of AIAA/AAS Astrodynamics Specialist Conferencec and Exhibit (2004)
22. Kuwata, Y., Balaram, J.B.: Combined EDL-mobility planning for planetary missions. In: Proceedings of Infotech@Aerospace (2011)
23. Kuwata, Y., Pavone, M., Balaram, J.B.: A risk-constrained multistage decision making approach to the architectural analysis of planetary missions. In: Proceedings of the IEEE Conference on Decision and Control (2012)
24. Li, P., Wendt, M., Wozny, G.: A probabilistically constrained model predictive controller. Automatica **38**, 1171–1176 (2002)
25. McGhan, C.L.R., Serra, R., Ingham, M.D., Ono, M., Estlin, T., Murray, R.M., Williams, B.C.: A risk-aware architecture for resilient spacecraft operations. In: Accepted for publication in the Proceedings of IEEE Aerospace Conference
26. Murray, R.M., Day, J.C., Ingham, M.D., Reder, L.J., Williams, B.C.: Engineering resilient space systems: Final report (2013)
27. National Aeronautics and Space Administration: Gravity Recovery and Interior Laboratory (GRAIL) Launch Press Kit
28. National Research Council of the National Academies: Vision and voyages for planetary science in the decade 2013-2022 (2010)
29. Nemirovski, A., Shapiro, A.: Convex approximations of chance constrained programs. SIAM Journal on Optimization **17**, 969–996 (2006)
30. Oldewurtel, F., Jones, C.N., Morari, M.: A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback. In: Proceedings of Conference on Decision and Control (2008)
31. Oldewurtel, F., Ulbig, A., Parisio, A., Andersson, G., Morari, M.: 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)
32. Ono, M.: Joint chance-constrained model predictive control with probabilistic resolvability. In: Proceedings of American Control Conference (2012)
33. Ono, M.: Robust, goal-directed plan execution with bounded risk. Ph.D. thesis, Massachusetts Institute of Technology (2012)
34. Ono, M., Topcu, U., Yo, M., Adachi, S.: Risk-limiting power grid control with an arma-based prediction model. In: Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, pp. 4949–4956 (2013)
35. Ono, M., Williams, B.C.: 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)
36. Parker, J.S., Anderson, R.L.: Low-Energy Lunar Trajectory Design. Wiley (2014)
37. Prakash, R., Burkhart, P., Chen, A., Comeaux, K., Guernsey, C., Kipp, D., Lorenzoni, L., Mendeck, G., Powell, R., Rivellini, T., San Martin, A., Sell, S., Steltzner, A., Way, D.: Mars science laboratory entry, descent, and landing system overview. In: Aerospace Conference, 2008 IEEE, pp. 1–18 (2008)
38. Rossman, L.A.: Reliability-constrained dynamic programing and randomized release rules in reservoir management. Water Resources Research (1977)
39. Schildbach, G., Goulart, P., Morari, M.: The linear quadratic regulator with chance constraints. In: Control Conference (ECC), 2013 European, pp. 2746–2751 (2013)
40. Schwarm, A.T., Nikolaou, M.: Chance-constrained model predictive control. AIChE Journal **45**(8), 1743–1752 (1999)
41. Sjödin, A.E., Gayme, D.F., Topcu, U.: Risk-mitigated optimal power flow with high wind penetration. In: Proceedings of American Control Conference (2012)
42. Summers, T., Warrington, J., Morari, M., Lygeros, J.: Stochastic optimal power flow based on convex approximations of chance constraints. In: Power Systems Computation Conference. Wroclaw, Poland (2014)
43. Vitus, M.P., Tomlin, C.J.: On feedback design and risk allocation in chance constrained control. In: Proc. IEEE Conf. on Decision and Control, pp. 734–739 (2011)

44. Williams, J.L., John W. Fisher, I., Willsky, A.S.: Approximate dynamic programming for communication-constrained sensor network management. IEEE Transactions on Signal Processing **55** (2007)
45. Yo, M., Ono, M., Williams, B., Adachi, S.: Risk-limiting, market-based power dispatch and pricing. In: European Control Conference, pp. 3038–3045 (2013)
46. Zhang, X., Grammatico, S., Schildbach, G., Goulart, P., Lygeros, J.: On the sample size of randomized MPC for chance-constrained systems with application to building climate control. In: European Control Conference, pp. 478–483. Strasbourg, France (2014)