Fast Approximate Dynamic Programming for Infinite-Horizon Markov Decision Processes

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the infinite-horizon, discounted cost, optimal control of discrete-time systems with separable cost and constraint in the state and input variables. Specifically, we introduce a novel numerical scheme for implementation of the value iteration (VI) algorithm in the conjugate domain, using the linear-time Legendre transform algorithm. Detailed analyses of the convergence, complexity, and error of the proposed algorithm are provided. In particular, with a discretization of size X and U for the state and input spaces, respectively, the time complexity of each iteration in the VI algorithm reduces from $\mathcal{O}(XU)$ to $\mathcal{O}(X+U)$.

1 Introduction

Value iteration (VI) is one of the most basic and wide-spread algorithms employed for tackling problems in reinforcement learning and optimal control [8, 25] formulated as Markov decision processes (MDPs). The VI algorithm simply involves the consecutive applications of the dynamic programming (DP) operator $\mathcal{T}V(x_t) = \min_{u_t} \left\{C(x_t, u_t) + \gamma \mathbb{E}V(x_{t+1})\right\}$, where $C(x_t, u_t)$ is the cost of taking the control action u_t at the state x_t. This fixed point iteration is known to converge to the optimal value function for discount factors $\gamma \in (0, 1)$. However, this algorithm suffers from a high computational cost for large scale finite state spaces. For problems with a continuous state space, the DP operation becomes an infinite-dimensional optimization problem, rendering the exact implementation of the VI algorithm impossible in most cases. A common approach is to incorporate function approximation techniques and compute the output of the DP operator for a finite sample (i.e., a discretization) of the underlying continuous state space. This approximation again suffers from a high computational cost for fine discretizations of the state space, particularly in high-dimensional problems. We refer the reader to [8, 23] for various schemes of approximate implementation of DP.

For some problems, however, it is possible to partially address this issue by using duality theory, i.e., approaching the minimization problem in the conjugate domain. In particular, as we will see in Section 3, the minimization in the primal domain in DP can be transformed to a simple addition in the dual domain, at the expense of three conjugate transforms. However, proper application of this transformation relies on efficient numerical algorithms for conjugation. Fortunately, such an algorithm, known as linear-time Legendre transform (LLT) has been developed in late 90s [20]. Other than the classical application of LLT (and other fast algorithms for conjugate transform) in solving Hamilton-Jacobi equation [1, 12, 13], these algorithms are used in image processing [21], thermodynamics [11], and optimal transport [16].

The application of conjugate duality for the DP problem is not new and actually goes back to Bellman [4]. Further applications of this idea for reducing the computational complexity were later explored in [14, 17]. However, surprisingly, the application of LLT for solving discrete-time optimal control problems has been limited. In particular, in [10], the authors propose the "fast value iteration" algorithm (without a rigorous analysis of the complexity and error of the proposed

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

algorithm) for a particular class of infinite-horizon optimal control problems with state-independent stage cost $C(x,u) = C(x)$ and linear dynamics $x_{t+1} = Ax_t + Bu_t$, where A is a non-negative, monotone, invertible matrix. More recently, in [18], the authors also considered the application of LLT for solving the DP operation in finite-horizon optimal control problems with a more general input-affine dynamics. In particular, they introduced the "discrete conjugate DP (d-CDP) operator," and provided a detailed analysis of its complexity and error. As we will discuss shortly, the current study is an extension of the corresponding d-CDP algorithm that, among other things, considers infinite horizon, discounted cost problems. We also note that the algorithms developed in [15, 21] for distance transform can also potentially tackle the optimal control problems similar to the ones of interest in the current study. However, these algorithms require the stage cost to be reformulated as a convex function of the "distance" between the current and next states, which can be restrictive. Another line of work that is closely related to ours invloves utilizing max-plus algebra in solving deterministic, continuous-state, continuous-time optimal control problems; see, e.g., [2, 22]. These works exploit the compatibility of the DP operation with max-plus operations, and approximate the value function as a max-plus linear combination. Recently, in [3, 5], the authors used this idea to propose an approximate value iteration algorithm for deterministic MDPs. In this regard, we note that the proposed approach in the current study also involves approximating the value function as a max-plus linear combination, namely, the maximum of a finite number of affine functions. The key difference is however that by choosing a grid-like (factorized) set of slopes for the linear terms (i.e., the basis of the max-plus linear combination), we take advantage of linear time complexity of LLT in computing the constant terms (i.e., the coefficients of the max-plus linear combination). Moreover, unlike previous works, we use the result of our error analysis to provide concrete guidelines on how to construct the basis dynamically at each iteration of the proposed VI algorithm.

Main contribution. In this study, we focus on the optimal control of discrete-time systems, with continuous state-input space. In particular, we incorporate an approximation of VI algorithm involving discretization of the state-input space. Building upon the earlier work [18], we employ conjugate duality to speed-up the VI algorithm for problems with separable stage cost (in state and input) and input-affine dynamics. We intorduce the conjugate VI (CVI) algorithm based on a modified version of the discrete conjugate DP operator introduced in [18], and extend the existing results for *infinite-horizon, discounted cost* problems, while taking into account *stochastic dynamics* and *numerical approximation of the conjugate of input-dependent stage cost* in our analysis. In particular, **(i)** we provide sufficient conditions for the convergence of the CVI algorithm (Theorem 3.9); **(ii)** we show that the CVI algorithm can achieve a linear time complexity of $\mathcal{O}(X+U)$ in each iteration (Theorem 3.10), where X and U are the cardinality the discrete state and input spaces, respectively; **(iii)** we analyze the error of the CVI algorithm (Theorem 3.11), and use that result to provide specific guidelines on the dynamic construction of discrete dual domain (Section 3.5).

Notations. The standard inner product in $\mathbb{R}^n$ and the corresponding induced 2-norm are denoted by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|_2$, respectively. $\|\cdot\|_\infty$ denotes the infinity norm. We use the superscript d to denote *finite (discrete)* sets (as in $\mathbb{X}^{\mathrm{d}}$) and *discrete* functions (as in $h^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$). We use the superscript g to denote *grid-like* finite sets (as in $\mathbb{X}^{\mathrm{g}} = \Pi_{i=1}^n \mathbb{X}_i^{\mathrm{g}}$). We also use $\mathbb{X}_{\mathrm{sub}}^{\mathrm{g}}$ to denote the *sub-grid* of $\mathbb{X}^{\mathrm{g}}$ derived by omitting the smallest and the largest elements of $\mathbb{X}^{\mathrm{g}}$ in each dimension. The cardinality of the finite set $\mathbb{X}^{\mathrm{d}}$ ($\mathbb{X}^{\mathrm{g}}$) is denoted by X. We use $\widetilde{h^{\mathrm{d}}} : \mathbb{R}^n \to \overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ to denote a generic *extension* of a discrete function h^{d}, and $\overline{h^{\mathrm{d}}} : \mathbb{R}^n \to \overline{\mathbb{R}}$ to denote *multilinear interpolation and extrapolation (LERP)* extension of a discrete function with a *grid-like* domain. Let $\mathbb{X}, \mathbb{Y}$ be two arbitrary sets in $\mathbb{R}^n$. $\mathrm{co}(\mathbb{X})$ is the convex hull of $\mathbb{X}$. We use $\mathrm{d}(\mathbb{X}, \mathbb{Y}) \coloneqq \inf_{x \in \mathbb{X}, y \in \mathbb{Y}} \|x - y\|_2$ to denote the distance between $\mathbb{X}$ and $\mathbb{Y}$. The one-sided Hausdorff distance *from* $\mathbb{X}$ *to* $\mathbb{Y}$ is defined as $\mathrm{d_H}(\mathbb{X}, \mathbb{Y}) \coloneqq \sup_{x \in \mathbb{X}} \inf_{y \in \mathbb{Y}} \|x - y\|_2$. Let $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ be an extended real-valued function with a non-empty effective domain $\mathrm{dom}(h) = \mathbb{X} \coloneqq \{x \in \mathbb{R}^n : h(x) < \infty\}$. The range of h is denoted by $\mathrm{Rng}(h) \coloneqq \max_{x \in \mathbb{X}} h(x) - \min_{x \in \mathbb{X}} h(x)$, and the subdifferential of h at a point $x \in \mathbb{X}$ is defined as $\partial h(x) \coloneqq \left\{ y \in \mathbb{R}^n : h(\tilde{x}) \geq h(x) + \langle y, \tilde{x} - x \rangle, \forall \tilde{x} \in \mathbb{X} \right\}$. We define $\mathbb{L}(h) \coloneqq \Pi_{i=1}^n \left[\mathrm{L}_i^-(h), \mathrm{L}_i^+(h) \right]$, where $\mathrm{L}_i^+(h)$ (resp. $\mathrm{L}_i^-(h)$) is the maximum (resp. minimum) slope of the function h along the i-th dimension We report the complexities using the standard big O notations $\mathcal{O}$ and $\widetilde{\mathcal{O}}$, where the latter hides the logarithmic factors. In this study, we are mainly concerned with the dependence of the computational complexities on *the size of the finite sets* involved (discretization of the primal and dual domains). In particular, we ignore the possible dependence of the computational complexities on the dimension of the variables, unless they appear in the power of the size of those discrete sets.

We note that the extended version of this article, including the technical proofs, is available in the supplementary material.

2 VI in primal domain

We are concerned with the infinite-horizon, discounted cost, optimal control problems of the form

$$V_\star(x) = \min\ \mathbb{E}_{w_t}\left[\sum_{t=0}^{\infty}\gamma^t C(x_t,u_t)\middle| x_0=x\right]$$
$$\text{s.t.}\ \ x_{t+1}=g(x_t,u_t,w_t),\ x_t\in\mathbb{X},\ u_t\in\mathbb{U},\ w_t\sim\mathbb{P}(\mathbb{W}),\quad \forall t\in\{0,1,\ldots\},$$

where $x_t\in\mathbb{R}^n$, $u_t\in\mathbb{R}^m$, and $w_t\in\mathbb{R}^l$ are the state, input and disturbance variables at time t, respectively; $\gamma\in(0,1)$ is the discount factor; $C:\mathbb{X}\times\mathbb{U}\to\mathbb{R}$ is the stage cost; $g:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^l\to\mathbb{R}^n$ describes the dynamics; $\mathbb{X}\subset\mathbb{R}^n$ and $\mathbb{U}\subset\mathbb{R}^m$ describe the state and input constraints, respectively; and, $\mathbb{P}(\cdot)$ is the distribution of the disturbance over the support $\mathbb{W}\subseteq\mathbb{R}^l$. Assuming the stage cost C is bounded, the optimal value function solves the Bellman equation $V_\star=\mathcal{T}V_\star$, where $\mathcal{T}$ is the corresponding DP operator (C and V are extended to infinity outside their effective domains) [6, Prop. 1.2.2]

$$\mathcal{T}V(x)\coloneqq\min_u\left\{C(x,u)+\gamma\cdot\mathbb{E}_w V\big(g(x,u,w)\big)\right\},\quad \forall x\in\mathbb{X}. \tag{1}$$

Indeed, the DP operator is γ-contractive in the infinity-norm [6, Prop. 1.2.4]. This property then gives rise to the VI algorithm $V_{k+1}=\mathcal{T}V_k,\ k=0,1,\ldots$, which converges to $V_\star$ as $k\to\infty$, for arbitrary initialization V_0. Moreover, assuming that the composition $V\circ g$ (for each w) and the cost C are jointly convex in state and input variables, the DP operator also preserves convexity [7, Prop. 3.3.1].

For numerical implementation of VI algorithm, we need to address three issues. First, we need to compute the expectation of the value function in (1). In order to simplify the exposition and include the computational cost of this operation explicitly, we focus on disturbances with finite support:

Assumption 2.1 (Disturbance with finite support)**.** *The disturbance w has a finite support $\mathbb{W}^{\mathrm{d}}\subset\mathbb{R}^l$ with a given probability mass function (p.m.f.) $p:\mathbb{W}^{\mathrm{d}}\to[0,1]$.*

Under the preceding assumption, we have $\mathbb{E}_w V\big(g(x,u,w)\big)=\sum_{w\in\mathbb{W}^{\mathrm{d}}}p(w)\cdot V\big(g(x,u,w)\big)$. The second and more important issue is that the optimization problem (1) is infinite-dimensional for the continuous state space $\mathbb{X}$. This renders the exact implementation of the VI algorithm impossible, except for a few cases with available closed form solution. A common solution to this problem is to deploy a sample-based approach, accompanied by a function approximation scheme. To be precise, for a finite subset $\mathbb{X}^{\mathrm{d}}$ of $\mathbb{X}$, at each iteration $k\geq 0$, we take the discrete function $V_k^{\mathrm{d}}:\mathbb{X}^{\mathrm{d}}\to\mathbb{R}$ as the input, and compute the discrete function $V_{k+1}^{\mathrm{d}}=[\mathcal{T}V_k^{\mathrm{d}}]^{\mathrm{d}}:\mathbb{X}^{\mathrm{d}}\to\mathbb{R}$, by employing an extension $\widetilde{V_k^{\mathrm{d}}}:\mathbb{X}\to\mathbb{R}$ of V_k^{d}, as an approximation of V_k in the DP operation (1). Finally, we have to actually solve the minimization problem over the control input in (1). Here, again, a common approximation involves enumeration over a finite subset $\mathbb{U}^{\mathrm{d}}$ of the inputs space $\mathbb{U}$. Incorporating these approximations, we end up with the *approximate* VI algorithm $V_{k+1}^{\mathrm{d}}=\mathcal{T}^{\mathrm{d}}V_k^{\mathrm{d}}$, characterized by the *discrete* DP (d-DP) operator

$$\mathcal{T}^{\mathrm{d}}V^{\mathrm{d}}(x)\coloneqq\min_{u\in\mathbb{U}^{\mathrm{d}}}\left\{C(x,u)+\gamma\cdot\textstyle\sum_{w\in\mathbb{W}^{\mathrm{d}}}p(w)\cdot\widetilde{V^{\mathrm{d}}}\big(g(x,u,w)\big)\right\},\quad \forall x\in\mathbb{X}^{\mathrm{d}}. \tag{2}$$

The convergence of the approximate VI algorithm described above depends on the properties of the extension operation $\widetilde{[\cdot]}$. In particular, if $\widetilde{[\cdot]}$ is non-expansive (in the infinity-norm), then $\mathcal{T}^{\mathrm{d}}$ is also γ-contractive. The error of the approximate VI algorithm also depends on the extension operation $\widetilde{[\cdot]}$ and its representative power. We refer the interested reader to [6, 9, 23] for detailed discussions on the convergence and error of approximate VI algorithm using different approximation schemes.

The d-DP operator and the corresponding approximate VI algorithm will be our benchmark for evaluating the performance of the alternative algorithm developed in this study. So, we finish this section with some remarks on the time complexity of the d-DP operation. Let the time complexity of a single evaluation of the extension operation $\widetilde{[\cdot]}$ be of $\mathcal{O}(E)$. Then, the time complexity of the d-DP operation (2) is of $\mathcal{O}(XUWE)$. In this regard, note that the scheme described above essentially

involves approximating a continuous-state/action MDP with a finite-state/action MDP, and then applying the VI algorithm. This, in turn, implies the lower bound $\Omega(XU)$ for the time complexity (corresponding to enumeration over $u \in \mathbb{U}^{\mathrm{d}}$ for each $x \in \mathbb{X}^{\mathrm{d}}$). This lower bound is also compatible with the best existing time complexities in the literature for VI for finite MDPs; see, e.g., [3, 24]. However, as we will see shortly, for a particular class of problems, it is possible to exploit the structure of the underlying continuous system in order to achieve a better time complexity.

3 Reducing complexity via conjugate duality

In this section, we introduce a class of problems that allow us to employ conjugate duality for the VI problem and propose an alternative path for implementing the corresponding DP operator. We also present the numerical scheme for implementing the proposed alternative path, and analyze its convergence, complexity, and error. We note that the proposed algorithm and its analysis are based on the d-CDP algorithm in [18, Sec. 5] for finite-horizon optimal control of deterministic systems. Here, we extend those results for infinite-horizon, discounted cost, optimal control of stochastic systems. Moreover, unlike [18], our analysis includes the case where the conjugate of input-dependent stage cost is not analytically available and has to be computed numerically; see also [18, Assump. 5.1].

3.1 Problem class

Throughout this section, we assume that the problem data satisfy the following conditions.

Assumption 3.1 (Problem class)**.** *The problem data has the following properties:* ***(i)*** *The* disturbance *is additive; that is, $g(x,u,w) = f(x,u) + w$, where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ describes the deterministic dynamics.* ***(ii)*** *The* deterministic dynamics *is of the form $f(x,u) = f_{\mathrm{s}}(x) + Bu$, where $f_{\mathrm{s}} : \mathbb{R}^n \to \mathbb{R}^n$ is a Lipschitz continuous, possibly nonlinear map describing the "state" dynamics, and $B \in \mathbb{R}^{n\times m}$.* ***(iii)*** *The* constraint *sets $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U} \subset \mathbb{R}^m$ are compact and convex. Moreover, for each $x \in \mathbb{X}$, the set of admissible inputs $\mathbb{U}(x) := \{u \in \mathbb{U} : g(x,u,w) \in \mathbb{X},\ \forall w \in \mathbb{W}^{\mathrm{d}}\}$ is nonempty.* ***(iv)*** *The* stage cost *C is separable in state and input, that is, $C(x,u) = C_{\mathrm{s}}(x) + C_{\mathrm{i}}(u)$, where $C_{\mathrm{s}} : \mathbb{X} \to \mathbb{R}$ and $C_{\mathrm{i}} : \mathbb{U} \to \mathbb{R}$ are Lipschitz continuous and convex.*

Some remarks are in order regarding the preceding assumptions. First, note that the setting of Assumption 3.1 goes beyond the classical LQR. In particular, it includes nonlinear state dynamics, state and input constraints, and non-quadratic stage costs. Second, the properties laid out in Assumption 3.1 imply that the set of admissible inputs $\mathbb{U}(x)$ is a compact set for each $x \in \mathbb{X}$. This, in turn, implies that the optimal value in (1) is achieved if $V : \mathbb{X} \to \mathbb{R}$ is also assumed to be lower semi-continuous. Hereafter, we also assume that the joint discretization of the state-input space is "proper" in the sense that the feasibility condition of Assumption 3.1-(iii) also holds for the discrete state-input space:

Assumption 3.2 (Feasibile discretization)**.** *The discrete state space $\mathbb{X}^{\mathrm{d}} \subset \mathbb{X}$ and input space $\mathbb{U}^{\mathrm{d}} \subset \mathbb{U}$ are such that $\mathbb{U}^{\mathrm{d}}(x) := \mathbb{U}(x) \cap \mathbb{U}^{\mathrm{d}} \neq \emptyset$ for all $x \in \mathbb{X}^{\mathrm{d}}$.*

3.2 VI in conjugate domain

In this subsection, we use duality theory to present an alternative path for computing the output of the DP operator through the conjugate domain. This path forms the basis for the algorithm proposed in this study. Fix $x \in \mathbb{X}$ and consider the following reformulation of the optimization problem (1)

$$\mathcal{T}V(x) = C_{\mathrm{s}}(x) + \min_{u,z} \left\{ C_{\mathrm{i}}(u) + \gamma \cdot \mathbb{E}_w V(z+w) : z = f(x,u) \right\},$$

where we used additivity of disturbance and separability of stage cost in Assumptions 3.1-(i,iv). The corresponding dual problem then reads as

$$\widehat{\mathcal{T}}V(x) := C_{\mathrm{s}}(x) + \max_{y} \min_{u,z} \left\{ C_{\mathrm{i}}(u) + \gamma \cdot \mathbb{E}_w V(z+w) + \langle y, f(x,u) - z \rangle \right\}, \tag{3}$$

where $y \in \mathbb{R}^n$ is dual variable corresponding to the equality constraint. For the dynamics of Assumption 3.1-(ii), we can obtain the following representation for the dual problem.

Proposition 3.3 (CDP operator)**.** *The dual problem* (3) *equivalently reads as*

$$\epsilon(x) := \gamma \cdot \mathbb{E}_w V(x+w), \quad x \in \mathbb{X}, \tag{4a}$$

$$\phi(y) := C_{\mathrm{i}}^*(-B^\top y) + \epsilon^*(y), \quad y \in \mathbb{R}^n, \tag{4b}$$

$$\widehat{\mathcal{T}}V(x) = C_{\mathrm{s}}(x) + \phi^*\big(f_{\mathrm{s}}(x)\big), \quad x \in \mathbb{X}, \tag{4c}$$

where $h^*(y) = \sup_x \{\langle y, x\rangle - h(x)\}$ is the (convex) conjugate of h.

Following [18], we call the operator $\widehat{\mathcal{T}}$ defined by (4) the *conjugate* DP (CDP) operator. We next provide an alternative representation of the CDP operator that captures the essence of this operation.

Proposition 3.4 (CDP reformulation)**.** *The CDP operator $\widehat{\mathcal{T}}$ equivalently reads as*

$$\widehat{\mathcal{T}}V(x) = C_{\mathrm{s}}(x) + \min_u \left\{C_{\mathrm{i}}(u) + \gamma \cdot [\mathbb{E}_w V(\cdot + w)]^{**}\big(f(x,u)\big)\right\}, \tag{5}$$

where $[\cdot]^{**} = [[\cdot]^*]^*$ denotes the biconjugate operation.

The preceding result implies that the indirect path through the conjugate domain essentially involves substituting the (expectation of the) value function by its biconjugate. In particular, it points to a sufficient condition for zero duality gap.

Corollary 3.5 (Equivalence of $\mathcal{T}$ and $\widehat{\mathcal{T}}$)**.** *If $V : \mathbb{X} \to \mathbb{R}$ is convex, then $\widehat{\mathcal{T}}V = \mathcal{T}V$.*

Hence, $\widehat{\mathcal{T}}$ has the same properties as $\mathcal{T}$ if V is convex. In particular, $\widehat{\mathcal{T}}$ is γ-contractive, and hence, the *conjugate* VI (CVI) algorithm $V_{k+1} = \widehat{\mathcal{T}}V_k,\ k = 0, 1, \ldots$, is expected to converge to the optimal value function $V_\star$ with arbitrary convex initialization V_0, if $\widehat{\mathcal{T}}$ preserves convexity (so that the output of each iteration of the CVI algorithm is convex). As before, this last condition holds if $V \circ f$ is jointly convex in x and u. This is for example the case for linear state dynamics $f_{\mathrm{s}}(x) = Ax,\ A \in \mathbb{R}^{n\times n}$. We note that, if $\widehat{\mathcal{T}}$ does not preserve convexity, then the alternative path suffers from duality gap.

3.3 CVI algorithm

The approximate CVI algorithm involves consecutive applications of an approximate implementation of the CDP operator (4) until some termination condition is satisfied. Algorithm 1 provides the pseudo-code of this procedure. In particular, we consider solving (4) for a finite set $\mathbb{X}^{\mathrm{d}}$ of points in the state space, and terminate the iterations when the difference between two consecutive discrete value functions (in the infinity-norm) is less than a given constant $e_{\mathrm{t}} > 0$; see Algorithm 1:4. In what follows, we describe the main steps within the initialization and the iterations of Algorithm 1. In particular, the three conjugate operations in (4) are handled numerically via the linear-time Legendre transform (LLT) algorithm [20]. LLT is an efficient algorithm for computing the *discrete* conjugate over a finite *grid-like* dual domain. Precisely, to compute the conjugate of the function $h : \mathbb{X} \to \mathbb{R}$, LLT takes its discretization $h^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$ as an input, and outputs $h^{\mathrm{d}*\mathrm{d}}(y) = [[h^{\mathrm{d}}]^*]^{\mathrm{d}}(y) = \max_{x\in\mathbb{X}^{\mathrm{d}}}\{\langle y, x\rangle - h^{\mathrm{d}}(x)\}$ for dual variables y belonging to a grid-like dual domain $\mathbb{Y}^{\mathrm{g}}$. We refer the reader to [20] for a detailed description of the LLT algorithm.

The main steps of the proposed CVI algorithm are as follows: **(i)** For the expectation operation in (4a), by Assumption 2.1, we again have $\mathbb{E}_w V(\cdot + w) = \sum_{w\in\mathbb{W}^{\mathrm{d}}} p(w) \cdot V(\cdot + w)$. Notice, however, that at each iteration, we only have the discrete value function $V^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$ at our disposal. Hence, we also need to employ some form of extension $\widetilde{V^{\mathrm{d}}} : \mathbb{X} \to \mathbb{R}$ of V^{d}. Therefore, we first pass the value function $V^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$ through the "scaled expection filter" to derive $\varepsilon^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$ in (6a), as an approximation of ϵ. **(ii)** In order to compute ϕ in (4b), we need access to two conjugate functions. First, for ϵ^*, we use the approximation $\varepsilon^{\mathrm{d}*\mathrm{d}} : \mathbb{Y}^{\mathrm{g}} \to \mathbb{R}$ in (6b), by applying LLT to the data points $\varepsilon^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$ for a properly chosen state dual grid $\mathbb{Y}^{\mathrm{g}} \subset \mathbb{R}^n$. We also need the conjugate C_{i}^* of the input-dependent stage cost. If this function is not analytically available, we approximate it as follows: For a properly chosen input dual grid $\mathbb{V}^{\mathrm{g}} \subset \mathbb{R}^m$, we employ LLT to compute $C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}} : \mathbb{V}^{\mathrm{g}} \to \mathbb{R}$ using the data points $C_{\mathrm{i}}^{\mathrm{d}} : \mathbb{U}^{\mathrm{d}} \to \mathbb{R}$; see (6c). With these conjugate functions at hand, we can now compute $\varphi^{\mathrm{d}} : \mathbb{Y}^{\mathrm{g}} \to \mathbb{R}$ in (6d), as an approximation of ϕ. In particular, notice that we use the LERP extension $\overline{C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}}}$ of $C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}}$ to approximate $C_{\mathrm{i}}^{\mathrm{d}*}$ at the required point $-B^\top y$ for each $y \in \mathbb{Y}^{\mathrm{g}}$. **(iii)** To be able to compute the output according to (4c), we need to perform another conjugate transform. In particular, we need the value of ϕ^* at $f_{\mathrm{s}}(x)$ for $x \in \mathbb{X}^{\mathrm{d}}$. Here, we use the approximation $\varphi^{\mathrm{d}*\mathrm{d}} : \mathbb{Z}^{\mathrm{g}} \to \mathbb{R}$ in (6e), by applying LLT to the data points $\varphi^{\mathrm{d}} : \mathbb{Y}^{\mathrm{g}} \to \mathbb{R}$ for a properly chosen grid $\mathbb{Z}^{\mathrm{g}} \subset \mathbb{R}^n$. Finally, we use the LERP extension $\overline{\varphi^{\mathrm{d}*\mathrm{d}}}$ of $\varphi^{\mathrm{d}*\mathrm{d}}$ to approximate $\varphi^{\mathrm{d}*}$ at the required point $f_{\mathrm{s}}(x)$ for each $x \in \mathbb{X}^{\mathrm{d}}$, and compute $\widehat{\mathcal{T}}^{\mathrm{d}}V^{\mathrm{d}}$ in (6f) as an approximation of $\widehat{\mathcal{T}}V$. With these approximations, we can

Algorithm 1 CVI: Approximate VI in conjugate domain

Input: dynamics $f_{\mathrm{s}} : \mathbb{R}^n \to \mathbb{R}^n$, $B \in \mathbb{R}^{n\times m}$; finite state space $\mathbb{X}^{\mathrm{d}} \subset \mathbb{X}$; finite input space $\mathbb{U}^{\mathrm{d}} \subset \mathbb{U}$; state-dependent cost function $C_{\mathrm{s}}^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$; input-dependent cost function $C_{\mathrm{i}}^{\mathrm{d}} : \mathbb{U}^{\mathrm{d}} \to \mathbb{R}$; finite disturbance space $\mathbb{W}^{\mathrm{d}}$ and its p.m.f. $p : \mathbb{W}^{\mathrm{d}} \to [0,1]$; discount factor γ; termination constant e_{t}.

Output: discrete value function $\widehat{V}^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$.

initialization:

construct the grid $\mathbb{V}^{\mathrm{g}}$; use LLT to compute $C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}} : \mathbb{V}^{\mathrm{g}} \to \mathbb{R}$ from $C_{\mathrm{i}}^{\mathrm{d}} : \mathbb{U}^{\mathrm{d}} \to \mathbb{R}$;

construct the grid $\mathbb{Z}^{\mathrm{g}}$;

$V^{\mathrm{d}}(x) \leftarrow 0$ for $x \in \mathbb{X}^{\mathrm{d}}$; $V_+^{\mathrm{d}}(x) \leftarrow C_{\mathrm{s}}^{\mathrm{d}}(x) - \min C_{\mathrm{i}}^{\mathrm{d}}$ for $x \in \mathbb{X}^{\mathrm{d}}$;

iteration:

while $\left\|V_+^{\mathrm{d}} - V^{\mathrm{d}}\right\|_\infty \geq e_{\mathrm{t}}$ **do**

  $V^{\mathrm{d}} \leftarrow V_+^{\mathrm{d}}$;

  construct the grid $\mathbb{Y}^{\mathrm{g}}$;

  d-CDP operation:

  $\varepsilon^{\mathrm{d}}(x) \leftarrow \gamma \cdot \sum_{w\in\mathbb{W}^{\mathrm{d}}} p(w)\widetilde{V^{\mathrm{d}}}(x+w)$ for $x \in \mathbb{X}^{\mathrm{d}}$; use LLT to compute $\varepsilon^{\mathrm{d}*\mathrm{d}} : \mathbb{Y}^{\mathrm{g}} \to \mathbb{R}$ from $\varepsilon^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$;

  for each $y \in \mathbb{Y}^{\mathrm{g}}$ **do**

    use LERP to compute $\overline{C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}}}(-B^\top y)$ from $C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}} : \mathbb{V}^{\mathrm{g}} \to \mathbb{R}$; $\varphi^{\mathrm{d}}(y) \leftarrow \overline{C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}}}(-B^\top y) + \varepsilon^{\mathrm{d}*\mathrm{d}}(y)$;

  end for

  use LLT to compute $\varphi^{\mathrm{d}*\mathrm{d}} : \mathbb{Z}^{\mathrm{g}} \to \mathbb{R}$ from $\varphi^{\mathrm{d}} : \mathbb{Y}^{\mathrm{g}} \to \mathbb{R}$;

  for each $x \in \mathbb{X}^{\mathrm{d}}$ **do**

    use LERP to compute $\overline{\varphi^{\mathrm{d}*\mathrm{d}}}\big(f_{\mathrm{s}}(x)\big)$ from $\varphi^{\mathrm{d}*\mathrm{d}} : \mathbb{Z}^{\mathrm{g}} \to \mathbb{R}$; $V_+^{\mathrm{d}}(x) \leftarrow C_{\mathrm{s}}(x) + \overline{\varphi^{\mathrm{d}*\mathrm{d}}}\big(f_{\mathrm{s}}(x)\big)$;

  end for

end while

output $\widehat{V}^{\mathrm{d}} \leftarrow V_+^{\mathrm{d}}$.

introduce the *discrete* CDP (d-CDP) operator as follows

$$\varepsilon^{\mathrm{d}}(x) \coloneqq \gamma \cdot \sum_{w\in\mathbb{W}^{\mathrm{d}}} p(w) \cdot \widetilde{V^{\mathrm{d}}}(x+w), \quad x \in \mathbb{X}^{\mathrm{d}}, \tag{6a}$$

$$\varepsilon^{\mathrm{d}*\mathrm{d}}(y) = \max_{x\in\mathbb{X}^{\mathrm{d}}} \left\{\langle x, y\rangle - \varepsilon^{\mathrm{d}}(u)\right\}, \quad y \in \mathbb{Y}^{\mathrm{g}}, \tag{6b}$$

$$C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}}(v) = \max_{u\in\mathbb{U}^{\mathrm{d}}} \left\{\langle u, v\rangle - C^{\mathrm{d}}(u)\right\}, \quad v \in \mathbb{V}^{\mathrm{g}}, \tag{6c}$$

$$\varphi^{\mathrm{d}}(y) \coloneqq \overline{C_{\mathrm{i}}^{\mathrm{d}*\mathrm{d}}}(-B^\top y) + \varepsilon^{\mathrm{d}*\mathrm{d}}(y), \quad y \in \mathbb{Y}^{\mathrm{g}}, \tag{6d}$$

$$\varphi^{\mathrm{d}*\mathrm{d}}(z) = \max_{y\in\mathbb{Y}^{\mathrm{g}}} \left\{\langle y, z\rangle - \varphi^{\mathrm{d}}(y)\right\}, \quad z \in \mathbb{Z}^{\mathrm{g}}, \tag{6e}$$

$$\widehat{\mathcal{T}}^{\mathrm{d}} V^{\mathrm{d}}(x) \coloneqq C_{\mathrm{s}}(x) + \overline{\varphi^{\mathrm{d}*\mathrm{d}}}\big(f_{\mathrm{s}}(x)\big), \quad x \in \mathbb{X}^{\mathrm{d}}. \tag{6f}$$

The proper construction of the grids $\mathbb{Y}^{\mathrm{g}}$, $\mathbb{V}^{\mathrm{g}}$, and $\mathbb{Z}^{\mathrm{g}}$ will be discussed in Section 3.5.

3.4 Analysis of CVI algorithm

We now provide our main theoretical results concerning the convergence, complexity, and error of the proposed algorithm. Before that, we present the assumptions to be called in this subsections.

Assumption 3.6 (Grids)**.** *Consider the following properties for the grids in Algorithm 1:* ***(i)*** *The gird $\mathbb{V}^{\mathrm{g}}$ is constructed such that $\mathrm{co}(\mathbb{V}_{\mathrm{sub}}^{\mathrm{g}}) \supseteq \mathbb{L}(C_{\mathrm{i}}^{\mathrm{d}})$;* ***(ii)*** *The grid $\mathbb{Z}^{\mathrm{g}}$ is constructed such that $\mathrm{co}(\mathbb{Z}^{\mathrm{g}}) \supseteq f_{\mathrm{s}}\big(\mathbb{X}^{\mathrm{d}}\big)$;* ***(iii)*** *The construction of $\mathbb{Y}^{\mathrm{g}}$, $\mathbb{V}^{\mathrm{g}}$, and $\mathbb{Z}^{\mathrm{g}}$ requires at most $\mathcal{O}(X+U)$ operations. $\mathbb{Y}^{\mathrm{g}}$ and $\mathbb{Z}^{\mathrm{g}}$ are of (approximately) the same size as $\mathbb{X}^{\mathrm{d}}$ in each dimension. $\mathbb{V}^{\mathrm{g}}$ is of (approximately) the same size as $\mathbb{U}^{\mathrm{d}}$ in each dimension.*

Assumption 3.7 (The extension operator)**.** *Consider the following properties for the extension operator $\widetilde{[\cdot]}$ in* (6a)*:* ***(i)*** *$\widetilde{[\cdot]}$ is non-expansive w.r.t. infinity norm;* ***(ii)*** *Given a function $V : \mathbb{X} \to \mathbb{R}$ and its discretization $V^{\mathrm{d}} : \mathbb{X}^{\mathrm{d}} \to \mathbb{R}$, we have $\|V - \widetilde{V^{\mathrm{d}}}\|_\infty \leq e_{\mathrm{e}}$ for some constant $e_{\mathrm{e}} \geq 0$.* ***(iii)*** *Each evaluation of $\widetilde{[\cdot]}$ has a time complexity of $\mathcal{O}(E)$.*

Assumption 3.8 (The dynamics)**.** *Given a convex function $V : \mathbb{X} \to R$, the composition $V \circ f$ is jointly convex in the state and input variables.*

We begin with the following result on the contractiveness of the d-CDP operator.

Theorem 3.9 (Convergence)**.** *Let Assumptions 3.6-(ii) and 3.7-(i) hold. Then, the d-CDP operator* (6) *is γ-contractive in the infinity-norm.*

The preceding theorem implies that the approximate CVI Algorithm 1 is indeed convergent given that the required conditions are satisfied. We next consider the time complexity of our algorithm.

Theorem 3.10 (Complexity)**.** *Let Assumptions 3.6-(iii) and 3.7-(iii) hold. The time complexity of initialization and each iteration in Algorithm 1 are of $\mathcal{O}(X+U)$ and $\widetilde{\mathcal{O}}(XWE+U)$, receptively.*

As a result, each iteration of the CVI algorithm requires at most $\widetilde{\mathcal{O}}(XWE+U)$ operations, where E denotes the complexity of the extension operator $\widetilde{[\cdot]}$ in (6a). On the other hand, recall that the time complexity of each iteration of the VI algorithm (in primal domain) is of $\mathcal{O}(XUWE)$, where E denotes the complexity of extension operation used in (2). In particular, for a deterministic dynamics ($W=1$), if $\mathbb{X}^{\mathrm{d}}=\mathbb{X}^{\mathrm{g}}$ is grid-like, and all the extension operations are handled using LERP (so that $E=\log X$), then the complexity of the CVI and VI algorithms are of $\widetilde{\mathcal{O}}(X+U)$ and $\widetilde{\mathcal{O}}(XU)$, respectively. This is a reduction from quadratic complexity to linear complexity. However, the CVI algorithm, like VI and other approximation schemes that utilize discretization/abstraction of the continuous state space, still suffers from the so-called "curse of dimensionality." This is because the size X of the discrete state space increases exponentially with the dimension of the state space. We finish with the following result on the error of the proposed CVI algorithm.

Theorem 3.11 (Error)**.** *Let Assumptions 3.6-(i,ii), 3.7-(i), and 3.8 hold. Consider the true optimal value function $V_\star=\mathcal{T}V_\star:\mathbb{X}\to\mathbb{R}$ and its discretization $V_\star^{\mathrm{d}}:\mathbb{X}^{\mathrm{d}}\to\mathbb{R}$. Let Assumption 3.7-(ii) hold for $V_\star$. Also, let $\widehat{V}^{\mathrm{d}}:\mathbb{X}^{\mathrm{d}}\to\mathbb{R}$ be the output of Algorithm 1 with* fixed *grids $\mathbb{Y}^{\mathrm{g}}$, $\mathbb{V}^{\mathrm{g}}$, and $\mathbb{Z}^{\mathrm{g}}$. Then,*

$$\left\|\widehat{V}^{\mathrm{d}}-V_\star^{\mathrm{d}}\right\|_\infty \le \frac{\gamma(e_{\mathrm{e}}+e_{\mathrm{t}})+e_{\mathrm{d}}}{1-\gamma}, \tag{7}$$

where $e_{\mathrm{d}}=e_{\mathrm{u}}+e_{\mathrm{v}}+e_{\mathrm{x}}+e_{\mathrm{y}}+e_{\mathrm{z}}$, and

$$\begin{aligned} &e_{\mathrm{u}}=c_{\mathrm{u}}\cdot \mathrm{d}_{\mathrm{H}}(\mathbb{U},\mathbb{U}^{\mathrm{d}}),\ e_{\mathrm{v}}=c_{\mathrm{v}}\cdot \mathrm{d}_{\mathrm{H}}\left(\mathrm{co}(\mathbb{V}^{\mathrm{g}}),\mathbb{V}^{\mathrm{g}}\right),\ e_{\mathrm{x}}=c_{\mathrm{x}}\cdot \mathrm{d}_{\mathrm{H}}\left(\mathbb{X},\mathbb{X}^{\mathrm{d}}\right),\\ &e_{\mathrm{y}}=c_{\mathrm{y}}\cdot \max\nolimits_{x\in\mathbb{X}^{\mathrm{d}}} \mathrm{d}\left(\partial(V_\star-C_{\mathrm{s}})(x),\mathbb{Y}^{\mathrm{g}}\right),\ e_{\mathrm{z}}=c_{\mathrm{z}}\cdot \mathrm{d}_{\mathrm{H}}\left(f_{\mathrm{s}}(\mathbb{X}^{\mathrm{d}}),\mathbb{Z}^{\mathrm{g}}\right), \end{aligned} \tag{8}$$

with constants $c_{\mathrm{u}},c_{\mathrm{v}},c_{\mathrm{x}},c_{\mathrm{y}},c_{\mathrm{z}}>0$ depending on the problem data.

The error terms in (7) relate to the main sources of error in the proposed approximate implementation of the CVI algorithm: **(i)** e_{e} is due to the approximation of the value function using the extension operator $\widetilde{[\cdot]}$; **(ii)** e_{t} corresponds to the termination of the algorithm after a finite number of iterations; **(iii)** e_{d} captures the error due to the discretization of the primal and dual state and input domains. We again note that Assumption 3.8 implies that the CDP operator preserves convexity and hence the duality gap is zero. Otherwise, the proposed scheme can suffer from large errors due to dualization.

3.5 Construction of the grids

In this final subsection, we provide specific guidelines for the construction of the grids $\mathbb{Y}^{\mathrm{g}}$, $\mathbb{V}^{\mathrm{g}}$ and $\mathbb{Z}^{\mathrm{g}}$. The presented guidelines are based on our error analysis (Theorem 3.11) and correspond to the properties laid out in Assumption 3.6. We refer the reader to [18] for further details about Assumption 3.6-(i,ii) on the grids $\mathbb{V}^{\mathrm{g}}$ and $\mathbb{Z}^{\mathrm{g}}$.

Static construction of $\mathbb{V}^{\mathrm{g}}$. By Assumption 3.6-(i), we need to find the smallest grid $\mathbb{V}^{\mathrm{g}}$ for which $\mathrm{co}(\mathbb{V}^{\mathrm{g}}_{\mathrm{sub}})\supseteq\mathbb{L}(C_{\mathrm{i}}^{\mathrm{d}})$, so that e_{v} in (8) is controlled by the resolution of $\mathbb{V}^{\mathrm{g}}$. This essentially means that $\mathbb{V}^{\mathrm{g}}$ must "more than cover the range of slopes" of the function $C_{\mathrm{i}}^{\mathrm{d}}$. Since C_{i} is convex (Assumption 3.1-(iv)), and hence $C_{\mathrm{i}}^{\mathrm{d}}$ is convex-extensible, we can compute the required range of slopes with an acceptable computational cost. In particular, assuming the domain $\mathbb{U}^{\mathrm{d}}=\mathbb{U}^{\mathrm{g}}=\Pi_{i=1}^m\mathbb{U}_i^{\mathrm{g}}$ of $C_{\mathrm{i}}^{\mathrm{d}}$ is grid-like, we can take $\mathrm{L}_i^-(C_{\mathrm{i}}^{\mathrm{d}})$ (resp. $\mathrm{L}_i^+(C_{\mathrm{i}}^{\mathrm{d}})$) to be the minimum first forward (resp. maximum last backward) difference of $C_{\mathrm{i}}^{\mathrm{d}}$ along the i-th dimension. We then construct $\mathbb{V}^{\mathrm{g}}_{\mathrm{sub}}=\Pi_{i=1}^m\mathbb{V}^{\mathrm{g}}_{\mathrm{sub}\,i}$ such that, in each dimension i, $\mathbb{V}^{\mathrm{g}}_{\mathrm{sub}\,i}$ is uniform with the same cardinality as $\mathbb{U}_i^{\mathrm{g}}$, and $\mathrm{co}(\mathbb{V}^{\mathrm{g}}_{\mathrm{sub}\,i})=\left[\mathrm{L}_i^-(C_{\mathrm{i}}^{\mathrm{d}}),\mathrm{L}_i^+(C_{\mathrm{i}}^{\mathrm{d}})\right]$. Finally, we construct $\mathbb{V}^{\mathrm{g}}$ by adding one smaller element and one greater element to $\mathbb{V}^{\mathrm{g}}_{\mathrm{sub}}$, while preserving the resolution, in each dimension. This construction satisfies the condition of Assumption 3.6-(iii).

Static construction of $\mathbb{Z}^{\mathrm{g}}$. According to Assumption 3.6-(ii), the grid $\mathbb{Z}^{\mathrm{g}}$ must be constructed such that $\mathrm{co}(\mathbb{Z}^{\mathrm{g}}) \supseteq f_{\mathrm{s}}(\mathbb{X}^{\mathrm{d}})$. This can be simply done by finding the vertices of the smallest box that contains the set $f_{\mathrm{s}}(\mathbb{X}^{\mathrm{d}})$. Those vertices give the range of $\mathbb{Z}^{\mathrm{g}}$ in each dimension. We can then, for example, take $\mathbb{Z}^{\mathrm{g}}$ to be the uniform grid with the same cardinality as $\mathbb{X}^{\mathrm{d}}$ in each dimension. This way, $\mathrm{d_H}\left(f_{\mathrm{s}}(\mathbb{X}^{\mathrm{d}}), \mathbb{Z}^{\mathrm{g}}\right) \leq \mathrm{d_H}\left(\mathrm{co}(\mathbb{Z}^{\mathrm{g}}), \mathbb{Z}^{\mathrm{g}}\right)$, and hence e_{z} in (8) is controlled by the resolution of $\mathbb{Z}^{\mathrm{g}}$. One again, this construction satisfies the complexity requirement of Assumption 3.6-(iii).

Static construction of $\mathbb{Y}^{\mathrm{g}}$. Construction of the dual state grid $\mathbb{Y}^{\mathrm{g}}$ is more involved. In particular, according to Theorem 3.11, we need to choose a *fixed* grid that minimizes e_{y} in (8). This can be done by choosing $\mathbb{Y}^{\mathrm{g}}$ such that $\mathbb{Y}^{\mathrm{g}} \cap \partial(V_\star - C_{\mathrm{s}}) \neq \emptyset$ for all $x \in \mathbb{X}^{\mathrm{g}}$ so that $e_{\mathrm{y}} = 0$. Even if we had access to the optimal value function $V_\star$, satisfying such a condition can lead to dual grids $\mathbb{Y}^{\mathrm{g}} \subset \mathbb{R}^n$ of size $\mathcal{O}(X^n)$. Such a large size violates Assumption 3.6-(iii) on the size of $\mathbb{Y}^{\mathrm{g}}$, and essentially renders the proposed algorithm impractical for dimensions $n \geq 2$. A more practical condition is $\mathrm{co}(\mathbb{Y}^{\mathrm{g}}) \cap \partial(V_\star - C_{\mathrm{s}}) \neq \emptyset$ for all $x \in \mathbb{X}^{\mathrm{d}}$ so that $\max_{x \in \mathbb{X}^{\mathrm{d}}} \mathrm{d}\left(\partial(V_\star - C_{\mathrm{s}})(x), \mathbb{Y}^{\mathrm{g}}\right) \leq \mathrm{d_H}\left(\mathrm{co}(\mathbb{Y}^{\mathrm{g}}), \mathbb{Y}^{\mathrm{g}}\right)$, and hence e_{y} is controlled by the resolution of the grid $\mathbb{Y}^{\mathrm{g}}$. In order to satisfy the latter condition, we need to approximate the range of slopes of $V_\star - C_{\mathrm{s}}$ (over $\mathbb{X}^{\mathrm{d}}$). To this end, we can use the fact that $V_\star$ is the fixed point of DP operator (1) to approximate $\mathrm{Rng}(V_\star - C_{\mathrm{s}})$ by $R = \frac{\mathrm{Rng}(C_{\mathrm{i}}^{\mathrm{d}}) + \gamma \cdot \mathrm{Rng}(C_{\mathrm{s}}^{\mathrm{d}})}{1-\gamma}$. We then construct the gird $\mathbb{Y}^{\mathrm{g}} = \Pi_{i=1}^{n} \mathbb{Y}_i^{\mathrm{g}}$ such that for each dimension $i = 1, \ldots, n$, we have $\pm {}^{\alpha R}/_{\Delta_{\mathbb{X}_i^{\mathrm{d}}}} \in \mathrm{co}(\mathbb{Y}_i^{\mathrm{g}})$, where $\Delta_{\mathbb{X}_i^{\mathrm{d}}}$ (with some abuse of notation) denotes the diameter of $\mathbb{X}^{\mathrm{d}}$ along i-th dimension, and $\alpha > 0$ is a scaling factor mainly depending on the dimension of the state space. This construction requires $\mathcal{O}(X + U)$ operations which is in consistence with Assumption 3.6-(iii).

Dynamic construction of $\mathbb{Y}^{\mathrm{g}}$. Alternatively, we can construct $\mathbb{Y}^{\mathrm{g}}$ *dynamically* at each iteration in order to minimize the corresponding error in each application of the d-CDP operator given by $e_{\mathrm{y}} = c_{\mathrm{y}} \cdot \max_{x \in \mathbb{X}^{\mathrm{d}}} \mathrm{d}\left(\partial(\mathcal{T}V - C_{\mathrm{s}})(x), \mathbb{Y}^{\mathrm{g}}\right)$; see Lemma 5.6 and Proposition 5.8 in the supplementary material. Once again, the idea is to make sure $\mathrm{co}(\mathbb{Y}^{\mathrm{g}}) \cap \partial(\mathcal{T}V - C_{\mathrm{s}}) \neq \emptyset$ for all $x \in \mathbb{X}^{\mathrm{g}}$. Since we do not have $[\mathcal{T}V]^{\mathrm{d}}$ at our disposal (it is the output of the current iteration), we can again use the definition of the DP operator (1) to approximate $\mathrm{Rng}(\mathcal{T}V - C_{\mathrm{s}})$ by $R = \mathrm{Rng}(C_{\mathrm{i}}^{\mathrm{d}}) + \gamma \cdot \mathrm{Rng}(V^{\mathrm{d}})$, where V^{d} is the output of the previous iteration. We can then construct $\mathbb{Y}^{\mathrm{g}}$ such that again for each dimension i, we have $\pm {}^{\alpha R}/_{\Delta_{\mathbb{X}_i^{\mathrm{d}}}} \in \mathrm{co}(\mathbb{Y}_i^{\mathrm{g}})$. This construction has a one-time computational cost of $\mathcal{O}(U)$ for computing $\mathrm{Rng}(C_{\mathrm{i}}^{\mathrm{d}})$, and per iteration computational cost of $\mathcal{O}(X)$ for computing $\mathrm{Rng}(V^{\mathrm{d}})$, which is again in consistence with Assumption 3.6-(iii).

4 Numerical simulations

We now showcase the application of the CVI algorithm in solving the optimal control problem of a noisy inverted pendulum, and compare its performance with the VI algorithm. The details of these simulations (and more numerical examples) are provided in Section 4 of the supplementary material. For these simulations, we use modified versions of multiple routines in the d-CDP MATLAB package [19]; see the supplementary material for the codes.

We use the setup (model and stage cost) of [18, App. C.2.2], with discount factor $\gamma = 0.95$ and termination constant $e_{\mathrm{t}} = 0.001$. In particular, the stage cost is quadratic (in both state and input), and the discrete-time dynamics is of the form $x^+ = f_{\mathrm{s}}(x) + Bu + w$, where $f_{\mathrm{s}}(x_1, x_2) = [x_1 + \alpha_{12} x_2,\ \alpha_{21} \sin x_1 + \alpha_{22} x_2]^\top$ is the *nonlinear* state dynamics, $B = [0, \beta]^\top$, and $w \in \mathbb{R}^2$ is the additive noise ($\alpha_{12}, \alpha_{21}, \alpha_{22}, \beta \in \mathbb{R}$). We use uniform, grid-like discretizations of the state and input spaces. This choice allows us to use LERP for the extension operators in (2) for VI and in (6a) for CVI. The grids $\mathbb{V}^{\mathrm{g}}$, $\mathbb{Z}^{\mathrm{g}}$, and $\mathbb{Y}^{\mathrm{g}}$ are also constructed uniformly, following the guidelines of Section 3.5 (with $\alpha = 1$). In particular, we consider both *static* and *dynamic* construction of the dual grid $\mathbb{Y}^{\mathrm{g}}$ in the CVI. Moreover, in each implementation of VI and CVI, all of the involved grids are chosen to be of the same size N_i in each dimension, i.e., $X = Y = Z = N_i^2$ and $U = V = N_i$. We are particularly interested in the performance of these algorithms, as the size of the discretizations increases. We note that with these grid sizes, the time complexity of each iteration of VI and CVI is of $\mathcal{O}(XUW)$ and $\mathcal{O}(XW + U)$, respectively, where $W = 5^2$ is the size of discrete disturbance set.

Let us also note that the optimal control problem at hand does not satisfy the feasibility condition of Assumption 3.2. Indeed, there exist $x \in \mathbb{X}^{\mathrm{g}}$ for which there is no $u \in \mathbb{U}^{\mathrm{g}}$ such that $x^+ =$

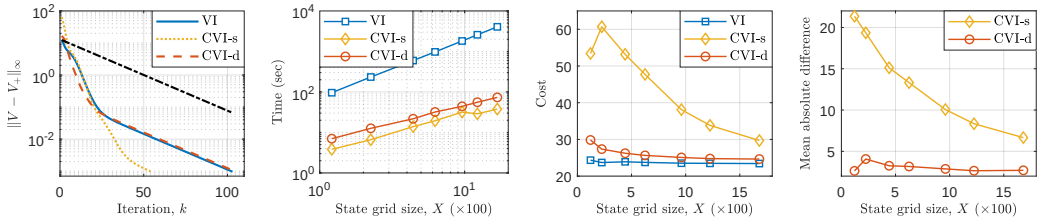

Figure 1: VI vs. CVI: (a) Convergence rate for grid size $X = 41^2$; (b) Running time; (c) Average cost of fifty instances of the control problem; (d) Difference between outputs of VI and CVI algorithms. The black dashed-dotted line in (a) corresponds to exponential convergence with rate $\gamma = 0.95$. CVI-s and CVI-d correspond to *static* and *dynamic* construction of the dual grid $\mathbb{Y}^{\mathrm{g}}$, respectively, in the CVI algorithm.

$f(x) + Bu \in \mathbb{X}$. Nevertheless, both algorithms can be used for computing the value function. However, the CVI algorithm, unlike the VI algorithm, assigns a finite value to the value function for these problematic states. This does not contradict our theoretical results on the error of the proposed algorithm, as the assumption on the optimal control problem to be feasible for all $x \in \mathbb{X}^{\mathrm{g}}$ is violated. Indeed, for the feasible initial states, our theoretical results still hold true. And, practically, we can simply impose the required state constraint while computing the control action in the forward iterations using the computed value function.

The results of our numerical simulations are shown in Figure 1. As shown in Figure 1a, both algorithms are indeed convergent with a rate greater than or equal to $\gamma = 0.95$. In particular, we see that CVI-s converges in $k_{\mathrm{t}} = 54$ iterations, compared to $k_{\mathrm{t}} = 105$ iterations required for CVI-d (with dynamic dual grid $\mathbb{Y}^{\mathrm{g}}$) and VI to reach the termination bound $e_{\mathrm{t}} = 0.001$. Our simulations show that the same increase in the convergence rate is also seen in CVI-d for *deterministic* systems; see Figure 2. (These effects were also seen for other grid sizes). On the other hand, the lower time complexity per iteration for CVI, leads to a significant reduction in the running time of this algorithm compared to VI. This effect can be clearly seen in Figure 1b, where the run-time of CVI for $X = 41^2$ is still less than the that of VI for $X = 11^2$. Moreover, as expected, using a fixed grid $\mathbb{Y}^{\mathrm{g}}$ in CVI-s, as opposed to dynamically constructing it at each iteration in CVI-d, leads to a reduced time requirement.

This reduction in the running time comes however with an increase in the cost of underlying optimal control problem. This effect is shown in Figure 1c, where we report the average cost of fifty instances of the optimal control problem with randomly chosen initial states, over $T = 400$ time steps. For these instances, at each time step, the control action is generated using the greedy policy $\mu(x) \in \operatorname{argmin}_{u \in \mathbb{U}^{\mathrm{g}}} C(x,u) + \gamma \cdot \overline{V^{\mathrm{d}}}\big(f(x,u)\big)$, w.r.t. the discrete value function V^{d} computed using VI and CVI. Notice that, compared to VI, we see a small increase in the cost of controlled trajectories in CVI-d, while using a static grid $\mathbb{Y}^{\mathrm{g}}$ in CVI-s causes a huge increase in the cost. This is because the dynamic construction of $\mathbb{Y}^{\mathrm{g}}$ in CVI uses the available computational power (related to size of the discretization) smartly by finding the smallest grid $\mathbb{Y}^{\mathrm{g}}$ in each iteration, in order to minimize the error of that same iteration. This effect is also seen in Figure 1d, where we report the mean absolute difference (over the discrete state space $\mathbb{X}^{\mathrm{g}}$) between the output of the CVI and VI algorithms. Once again, using a dynamic dual grid $\mathbb{Y}^{\mathrm{g}}$ leads to less difference between the two algorithm, which in turn implies smaller errors.

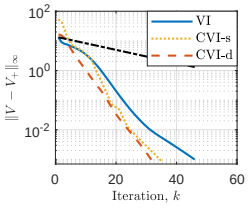

Figure 2: Convergence rate for grid size $X = 41^2$ and *deterministic* dynamics $x^+ = f_{\mathrm{s}}(x) + Bu$; see also Figure 1.

References

[1] Y. Achdou, F. Camilli, and L. Corrias. On numerical approximation of the Hamilton-Jacobi-transport system arising in high frequency approximations. *Discrete & Continuous Dynamical Systems-Series B*, 19(3), 2014.

[2] M. Akian, S. Gaubert, and A. Lakhoua. The max-plus finite element method for solving deterministic

optimal control problems: Basic properties and convergence analysis. *SIAM Journal on Control and Optimization*, 47(2):817–848, 2008.

[3] F. Bach. Max-plus matching pursuit for deterministic Markov decision processes. *arXiv preprint arXiv:1906.08524*, 2019.

[4] R. Bellman and W. Karush. Mathematical programming and the maximum transform. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):550–567, 1962.

[5] E. Berthier and F. Bach. Max-plus linear approximations for deterministic continuous-state markov decision processes. *IEEE Control Systems Letters*, pages 1–1, 2020.

[6] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, Belmont, MA, 3rd edition, 2007.

[7] D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, Belmont, MA, 2009.

[8] D. P. Bertsekas. *Reinforcement Learning and Optimal Control*. Athena Scientific, Belmont, MA, 2019.

[9] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst. *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2017.

[10] R. Carpio and T. Kamihigashi. Fast value iteration: an application of Legendre-Fenchel duality to a class of deterministic dynamic programming problems in discrete time. *Journal of Difference Equations and Applications*, 26(2):209–222, 2020.

[11] L. Contento, A. Ern, and R. Vermiglio. A linear-time approximate convex envelope algorithm using the double Legendre–Fenchel transform with application to phase separation. *Computational Optimization and Applications*, 60(1):231–261, 2015.

[12] L. Corrias. Fast Legendre-Fenchel transform and applications to Hamilton-Jacobi equations and conservation laws. *SIAM Journal on Numerical Analysis*, 33(4):1534–1558, 1996.

[13] G. Costeseque and J.-P. Lebacque. A variational formulation for higher order macroscopic traffic flow models: Numerical investigation. *Transportation Research Part B: Methodological*, 70:112 – 133, 2014.

[14] A. O. Esogbue and C. W. Ahn. Computational experiments with a class of dynamic programming algorithms of higher dimensions. *Computers & Mathematics with Applications*, 19(11):3 – 23, 1990.

[15] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. *Theory of computing*, 8(1):415–428, 2012.

[16] M. Jacobs and F. Léger. A fast approach to optimal transport: the back-and-forth method. *arXiv preprint arXiv:1905.12154*, 2019.

[17] C. M. Klein and T. L. Morin. Conjugate duality and the curse of dimensionality. *European Journal of Operational Research*, 50(2):220 – 228, 1991.

[18] M. A. S. Kolarijani and P. Mohajerin Esfahani. Fast approximate dynamic programming for input-affine dynamics. *preprint arXiv:2008.10362*, 2020.

[19] M. A. S. Kolarijani and P. Mohajerin Esfahani. Discrete conjugate dynamic programming (d-CDP) MATLAB package. Available online at `https://github.com/AminKolarijani/d-CDP`, 2020.

[20] Y. Lucet. Faster than the fast Legendre transform, the linear-time Legendre transform. *Numerical Algorithms*, 16(2):171–185, 1997.

[21] Y. Lucet. New sequential exact Euclidean distance transform algorithms based on convex analysis. *Image and Vision Computing*, 27(1):37 – 44, 2009.

[22] W. M. McEneaney. Max-plus eigenvector representations for solution of nonlinear H_∞ problems: basic concepts. *IEEE Transactions on Automatic Control*, 48(7):1150–1163, 2003.

[23] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, Hoboken, NJ, 2nd edition, 2011.

[24] A. Sidford, M. Wang, X. Wu, and Y. Ye. Variance reduced value iteration and faster algorithms for solving Markov decision processes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 770–787. SIAM, 2018.

[25] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Theorems 3.9, 3.10 and 3.11.

(b) Did you describe the limitations of your work? [Yes] See the discussions following Theorems 3.10 and 3.11.
(c) Did you discuss any potential negative societal impacts of your work? [N/A]
(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 5 of the supplementary material.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4 of the supplementary material.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See [19].
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]