On Distributionally Robust Multistage Convex Optimization: New Algorithms and Complexity Analysis

Shixuan Zhang and Xu Andy Sun

October 13, 2020

Abstract

This paper presents a novel algorithmic study and complexity analysis of distributionally robust multistage convex optimization (DR-MCO). We propose a new class of algorithms for solving DR-MCO, namely a sequential dual dynamic programming (Seq-DDP) algorithm and its nonsequential version (NDDP). The new algorithms generalize and strengthen existing DDP-type algorithms by introducing the technique of regularization that enables the algorithms to handle 1) fast growth of Lipschitz constants, which is a common phenomenon in multistage optimization, and 2) problems without relatively complete recourse. We then provide a thorough complexity analysis of the new algorithms, proving both upper complexity bounds and a matching lower bound. To the best of our knowledge, this is the first complexity analysis of DDP-type algorithms for DR-MCO problems, quantifying the dependence of the oracle complexity of DDP-type algorithms on the number of stages, the dimension of the decision space, and various regularity characteristics of DR-MCO. Numerical examples are also given to show the effectiveness of the proposed regularization technique on reducing computation time or the number of oracle evaluations and on solving problems without relatively complete recourse.

Keywords: distributionally robust optimization; multistage convex optimization; dual dynamic programming algorithm; complexity analysis; cutting plane algorithm

1 Introduction

Distributionally robust multistage convex optimization (DR-MCO) is a sequential decision making problem with convex objective functions and constraints, where the exact probability distribution of the uncertain parameters is unknown and decisions need to be made considering a family of distributions. DR-MCO provides a unified framework for studying decision-making under uncertainty. It includes as special cases both multistage stochastic convex optimization (MSCO), where the distribution of uncertainty is known and an average solution is sought, and multistage robust convex optimization (MRCO), where the uncertainty is described by a set and a worst-case solution is sought. DR-MCO as a general decision framework finds ubiquitous applications in energy system planning, supply

chain and inventory control, portfolio optimization, finance, and many other areas (see e.g. [31, 5]).

DR-MCO is in general challenging to solve, due to the fast growth of the number of decisions with respect to the number of decision stages [29, 13, 15]. Meanwhile, real-world problems are often endowed with special structures in the uncertainty distribution. In particular, the uncertainty may exhibit stagewise independence (SI), i.e. the joint probability distribution equals the product of its marginal distributions in each stage. Many uncertainty structures, such as autoregressive models, can be reformulated into SI. This versatile modeling capability of SI has great implications on computation. It allows recursive formulation of a cost-to-go function in each stage of a DR-MCO to be independent of the outcomes in its parent stages, thus making efficient approximations of the cost-to-go functions possible. Indeed, SI has been successfully exploited by various dual dynamic programming algorithms in solving MSCO and MRCO [30, 28, 15, 26, 3, 4, 1, 35, 32].

Dual dynamic programming (DDP) is a class of recursive cutting plane algorithms that originate from nested Benders decomposition for multistage stochastic linear optimization [7, 24]. The earliest form of DDP for MSCO using stochastic sampling method was proposed in [25], where in each iteration the scenarios are sampled randomly and solved sequentially before updating the cost-to-go functions recursively. The algorithm has since been widely adopted in areas such as energy systems scheduling [14, 9, 33]. The deterministic sampling version of DDP was later proposed, which uses both over- and under-approximations for sampling and termination [3, 4]. DDP has also been extended to multistage stochastic nonconvex problems [35, 36, 1, 34]. Robust dual dynamic programming (RDDP) is proposed for multistage robust linear optimization [15]. Due to its intrinsic difficulty, the uncertainty sets are assumed to be polytopes such that the subproblem in each stage can be solved via a vertex enumeration technique over the uncertainty set. Similar to the deterministic DDP, RDDP constructs both over- and under-approximations to select the worst-case outcome. Moreover, it has the advantage of being able to terminate the algorithm with a guaranteed optimal first stage solution, in contrast to the commonly used decision rules [20, 6, 18]. Recently, DDP has been further extended to DR-MCO with promising out-of-sample performance [2, 19, 27, 12]. In particular, [27] uses ambiguity set defined by a χ^2-distance neighborhood. In [12], the ambiguity sets are taken to be finitely supported Wasserstein metric balls centered at the empirical distributions, and the algorithm is shown to converge asymptotically with stochastic sampling methods. We comment that all of the above variants of DDP algorithms rely on the assumption of relatively complete recourse, while it is indeed possible to have MSCO without such assumption [23].

The convergence analysis of DDP begins with multistage linear optimization [28, 30, 22, 8], where an almost sure finite convergence is established based on polyhedral structures. In [16], an asymptotic convergence is proved for MSCO problems. Due to the multistage structure, a main complexity question concerning DDP is the dependence of its iteration complexity on the number of decision stages, which is recently answered for MSCO, independently in [21, 34]. In particular, [16, 21] assume relatively complete recourse and, moreover, the value functions are all Lipschitz continuous. A weaker regularity assumption

concerning exact penalization is made instead in [34] that allows the DDP algorithm to work without relatively complete recourse. However, it is not yet known whether the DDP algorithm, together with the complexity analysis, works for DR-MCO. The present paper introduces new algorithmic techniques to DDP and answers this question for the first time.

While the DDP algorithms have achieved great computational success for MSCO problems, one major issue of their generalization to the DR-MCO problems is the lack of proper termination criterion. Due to the distributional uncertainty in the model, the commonly used statistical upper bound for the policy evaluation in the MSCO literature (e.g., [30, 35]) is no longer valid for the DR-MCO problems. As a result, the current computational implementations in [27] and [12] choose to terminate at a fixed number of iterations or cuts, without a good guarantee of the solution quality. To overcome the lack of statistical upper bound, we focus on the deterministic DDP algorithms in this paper, which use both under- and over-approximations to certify the optimality of the solution, similar to the RDDP algorithm [15]. In particular, our paper makes the following contributions.

1. We provide a unified framework for studying DR-MCO with finitely supported distributions. Under the framework, we construct a novel example to show that the traditional cutting plane method can easily cause the Lipschitz constants of the stage problem to grow with respect to the number of stages. Motivated by this phenomenon, we introduce an important algorithmic technique of regularization to DDP, which can effectively control the growth of Lipschitz constraints and can dispense with the relatively complete recourse assumption.
2. A new class of sequential DDP algorithms (Seq-DDP) and its nonsequential version (NDDP) based on regularization is proposed for solving DR-MCO. Complexity upper bounds based on single stage subproblem oracles are proved for both Seq-DDP and NDDP for the first time.
3. We construct a class of multistage robust convex problems to obtain a complexity lower bound for the new algorithms for the first time, which shows the complexity upper bounds are essentially tight, in terms of the number of stages. The complexity bounds may be applied to more general DR-MCO problems with continuous distributions.
4. Numerical results on a multi-commodity inventory problem and a hydro-thermal power planning problem are given to illustrate the two effects of regularization: capability to solve problems without relatively complete recourse and reduction in the computation time and number of subproblem oracle evaluations.

The rest of the paper is organized as follows. Section 2 contains the formulations of the problems and the discussion on the regularization. In Section 3, we introduce the single stage subproblem oracles and present the algorithms with complexity analyses (both upper bounds and a lower bound). In Section 4, we present two classes of numerical examples that demonstrate the effectiveness of the algorithms using the regularization technique. We provide some concluding remarks in Section 5.

2 Formulations and Recursive Approximation

In this section, we introduce formulations of distributionally robust multistage convex optimization (DR-MCO) with finite supports. We build approximations of the value functions using recursions. We then discuss the regularization technique and its exactness for complexity analysis in Section 3.

2.1 Problem Formulations

We first describe the standard formulation for DR-MCO and then show that a general class of multistage robust convex optimization (MRCO) problems can be rewritten in the standard formulation.

2.1.1 Distributionally Robust Multistage Stochastic Convex Optimization

We begin with the definition of distributionally robust multistage convex optimization (DR-MCO) problems that are stagewise independent (SI) and have finite supports. Let $\mathcal{T} \coloneqq \{1, 2, \ldots, T\}$ denote the set of stage indices and $\mathcal{N} = \cup_{t=1}^{T} \mathcal{N}(t)$ denote the set of subproblem indices, which is partitioned by the stages. In particular, the first stage is assumed to be deterministic, i.e., $\mathcal{N}(1) = \{1\}$ is a singleton. The pair $(\mathcal{T}, \mathcal{N})$ indeed defines a recombining scenario tree for stagewise independent stochastic programs [34, 35]. For each subproblem $n \in \mathcal{N}$, let $\mathcal{F}_n$ denote the compact feasibility set of decision variables (x_n, y_n) in some Euclidean spaces, where x_n is called the state variable and y_n the internal variable. The state space $\mathcal{X}_t \subset \mathbb{R}^{d_t}$ of stage $t \in \mathcal{T}$ is a compact subset that contains all the projections of feasibility sets $\mathcal{F}_n$ onto the state variable for $n \in \mathcal{N}(t)$. We make the convention $x_0 \in \mathcal{X}_0 = \{0\}$ and $\mathcal{N}(T+1) = \varnothing$ for notational convenience. The cost function of the subproblem $n \in \mathcal{N}(t)$ is a nonnegative lower semicontinuous function $f_n(x_{t-1}, y_n, x_n)$, which is allowed to take $+\infty$ as an indicator function, for all $t \in \mathcal{T}$. The DR-MCO is defined as follows.

$$\begin{aligned} \min_{(x_1,y_1)\in\mathcal{F}_1,} \quad f_1(x_0, y_1, x_1) + \sup_{p_1\in\mathcal{P}_1} \sum_{n_2\in\mathcal{N}(2)} p_{1,n_2} \cdot \min_{(x_{n_2},y_{n_2})\in\mathcal{F}_{n_2}} \Bigg[f_{n_2}(x_1, y_{n_2}, x_{n_2}) + \\ + \sup_{p_{n_2}\in\mathcal{P}_2} \sum_{n_3\in\mathcal{N}(3)} p_{n_2,n_3} \cdot \min_{(x_{n_3},y_{n_3})\in\mathcal{F}_{n_3}} \Big[f_{n_3}(x_{n_2}, y_{n_3}, x_{n_3}) + \cdots \\ + \sup_{p_{n_T}\in\mathcal{P}_T} \sum_{n_T\in\mathcal{N}(T)} p_{n_{T-1},n_T} \cdot \min_{(x_{n_T},y_{n_T})\in\mathcal{F}_{n_T}} f_{n_T}(x_{n_{T-1}}, y_{n_T}, x_{n_T}) \Big] \Bigg]. \end{aligned} \tag{1}$$

In this definition (1), each transition probability vector $p_n = (p_{n,m})_{m\in\mathcal{N}(t+1)}$ belongs to a family of discrete distributions $\mathcal{P}_t$ for $n \in \mathcal{N}(t)$. Note that the set $\mathcal{P}_t$ depends only on its stage, regardless of uncertainties and realizations in its previous stages. This is known as the stagewise independence for multistage stochastic and robust programs [30, 15]. We assume that the problem (1) has a finite optimal objective value such that it is well-defined.

By the introduction of (worst-case) expected cost-to-go function, we are able to rewrite the problem (1) as a recursion:

$$\mathcal{Q}_t(x_t) \coloneqq \sup_{p_t \in \mathcal{P}_t} \sum_{n \in \mathcal{N}(t+1)} p_{t,n} \left(\min_{(x_n, y_n) \in \mathcal{F}_n} f_n(x_t, y_n, x_n) + \mathcal{Q}_{t+1}(x_n) \right). \tag{2}$$

By convention, $\mathcal{Q}_T(x) \equiv 0$. To simply the notation, we also define the value functions as

$$Q_n(x_t) \coloneqq \min_{(x_n, y_n) \in \mathcal{F}_n} f_n(x_t, y_n, x_n) + \mathcal{Q}_{t+1}(x_n), \quad n \in \mathcal{N}(t+1). \tag{3}$$

We have the obvious relation $\mathcal{Q}_t(x_t) = \sup_{p_t \in \mathcal{P}_t} \sum_{n \in \mathcal{N}(t+1)} p_{t,n} Q_n(x_t)$ for all $t < T$. Since the size of the problem (1) may grow exponentially with the number of stages T, it is often impractical to seek subproblem solutions (x_n, y_n) for all $n \in \mathcal{N}$. Our goal is to solve instead for an optimal first stage solution (x_1, y_1) to the problem (1), which can be equivalently written as

$$(x_1^*, y_1^*) \in \operatorname*{arg\,min}_{(x_1, y_1) \in \mathcal{F}_1} \left\{ f_1(x_0, y_1, x_1) + \mathcal{Q}_1(x_1) \right\}. \tag{4}$$

One important observation from the recursive formulation (2) is that the supremum is defined over a linear function in the probability vector p_t. Therefore, we assume that $\mathcal{P}_t \subseteq \Delta^{|\mathcal{N}(t+1)|} \coloneqq \{p_t \in \mathbb{R}_{\geq 0}^{|\mathcal{N}(t+1)|} : \sum_{n \in \mathcal{N}(t+1)} p_{t,n} = 1\}$ is a closed convex subset without loss of generality. Thus we replace the supremum in definition (2) with maximum in the rest of the paper. The following proposition characterizes the convexity of the recursion functions (2) and (3).

Proposition 1. *If the feasibility sets $\mathcal{F}_n$ are compact convex, and local cost functions f_n are closed convex for all $n \in \mathcal{N}$, then the functions Q_n and $\mathcal{Q}_t$ are closed convex for all $n \in \mathcal{N}$ and $t \in \mathcal{T}$.*

Proof. We prove by recursion from $t = T$ to $t = 1$. By definition, $\mathcal{Q}_T \equiv 0$ is closed convex. Now assume $\mathcal{Q}_t$ is closed convex for some $t \in \mathcal{T}$. For each $n \in \mathcal{N}(t)$, by assumption $f_n(x_{t-1}, y_n, x_n) + \mathcal{Q}_t(x_n)$ is a closed convex function on $\mathcal{F}_n$. Therefore, the epigraph of $Q_n(x_{t-1})$ on $\mathcal{X}_{t-1}$ is closed convex and so is Q_n by definition (3). For any probability vector $p_{t-1} \in \mathcal{P}_{t-1}$, the epigraph of the weighted sum $\sum_{n \in \mathcal{N}(t)} p_{t-1} Q_n(x_{t-1})$ is a closed convex set. The epigraph of $\mathcal{Q}_{t-1}$ is the intersection of these epigraphs, which shows that $\mathcal{Q}_{t-1}$ is a closed convex function. □

2.1.2 Multistage Robust Convex Optimization

Another important source of our formulation is the multistage robust convex optimization (MRCO) problems, where the uncertainty sets are polytopes. In this section, we describe these robust programs and show that they can be equivalently formulated as the problem (2).

Let $\mathcal{T}$ denote the set of stage indices. For each stage $t \in \mathcal{T}$, let ξ_t denote the uncertainty vector taking values from an uncertainty set Ξ_t. The first stage is assumed to be certain,

i.e., Ξ_1 is a singleton and, without loss of generality, we write $\xi_1 \equiv 0$. Let (x_t, y_t) denote the decision variables such that the tuple $(x_t, y_t; \xi_t)$ is restricted in the feasibility set $\mathcal{F}_t^{\text{rob}}$, and $f_t^{\text{rob}}(x_{t-1}, y_t, x_t; \xi_t)$ the local cost function for each stage $t \in \mathcal{T}$, with the convention $x_0 \equiv 0$. The MRCO can be written as

$$\begin{aligned} \min_{(x_1,y_1;\xi_1)\in\mathcal{F}_1^{\text{rob}}} \quad & f_1^{\text{rob}}(x_0, y_1, x_1; \xi_1) \; + \sup_{\xi_2\in\Xi_2} \min_{(x_2,y_2;\xi_2)\in\mathcal{F}_2^{\text{rob}}} f_2^{\text{rob}}(x_1, y_2, x_2; \xi_2) \\ & + \sup_{\xi_3\in\Xi_3} \min_{(x_3,y_3;\xi_3)\in\mathcal{F}_3^{\text{rob}}} f_3^{\text{rob}}(x_2, y_3, x_3; \xi_3) + \cdots \\ & + \sup_{\xi_T\in\Xi_T} \min_{(x_T,y_T;\xi_T)\in\mathcal{F}_T^{\text{rob}}} f_T^{\text{rob}}(x_{T-1}, y_T, x_T; \xi_T). \end{aligned} \tag{5}$$

We similarly define the (worst-case) cost-to-go functions in a recursive fashion.

$$\mathcal{Q}_{t-1}(x_{t-1}) \coloneqq \sup_{\xi_t\in\Xi_t} \min_{x_t,y_t} \left\{ f_t^{\text{rob}}(x_{t-1}, y_t, x_t; \xi_t) + \mathcal{Q}_t(x_t) : (x_t, y_t; \xi_t) \in \mathcal{F}_t^{\text{rob}} \right\}, \tag{6}$$

where $\mathcal{Q}_T \equiv 0$. The goal of the MRCO is to find a first stage optimal solution

$$(x_1^*, y_1^*) \in \operatorname*{arg\,min}_{(x_1,y_1;\xi_1)\in\mathcal{F}_1^{\text{rob}}} \left\{ f_1^{\text{rob}}(x_0, y_1, x_1; \xi_1) + \mathcal{Q}_1(x_1) \right\}. \tag{7}$$

The recursion (6) is in general very challenging to solve because it involves finding the supremum over a usually nonconcave function in the uncertainty vectors ξ_t. However, in an important class of problem described below, the recursion (6) can be equivalently reformulated as equation (2).

Proposition 2. *If the feasibility sets $\mathcal{F}_t^{\text{rob}}$ are convex, the uncertainty sets Ξ_t are polytopes, and the local cost functions f_t^{rob} are jointly convex in $(x_t, y_t; \xi_t)$ for all $t \in \mathcal{T}$, then*

$$\mathcal{Q}_t(x_t) = \max_{p_t\in\Delta^N} \sum_{n\in\mathcal{N}(t+1)} p_{t,n} \left(\min_{(x_n,y_n)\in\mathcal{F}_n} f_n(x_t, y_n, x_n) + \mathcal{Q}_{t+1}(x_n) \right),$$

where $\operatorname{ext}\Xi_{t+1} =: \{\xi_n\}_{n\in\mathcal{N}(t+1)}$ *is the finite set of extreme points of* Ξ_{t+1}, $N \coloneqq |\mathcal{N}(t+1)|$, Δ^N *the N-dimensional simplex, and*

$$f_n(x_t, y_n, x_n) \coloneqq f_{t+1}^{\text{rob}}(x_t, y_n, x_n; \xi_n), \quad \mathcal{F}_n \coloneqq \{(x_n, y_n) : (x_n, y_n; \xi_n) \in \mathcal{F}_{t+1}^{\text{rob}}\}.$$

Proof. By definition (6) and the same argument in Proposition 1, each cost-to-go function $\mathcal{Q}_t$ is convex if the local cost functions f_t^{rob} and feasibility sets $\mathcal{F}_t^{\text{rob}}$ are convex for all $t \in \mathcal{T}$. Then the function $Q_{t-1}^{\text{rob}}(x_{t-1}; \xi_t) \coloneqq \min_{x_t,y_t}\{f_t^{\text{rob}}(x_{t-1}, y_t, x_t; \xi_t) + \mathcal{Q}_t(x+t)\}$ is also convex. Note that $\mathcal{Q}_{t-1}(x_{t-1}) = \sup_{\xi_t\in\Xi_t} Q_t^{\text{rob}}(x_{t-1}; \xi_t)$. By convexity, we conclude that $\mathcal{Q}_{t-1}(x_{t-1}) = \sup_{\xi_t\in\operatorname{ext}\Xi_t} Q_t^{\text{rob}}(x_{t-1}; \xi_t) = \max_{p_{t-1}\in\Delta^N} \sum_{n\in\mathcal{N}(t)} p_{t-1,n} Q_t^{\text{rob}}(x_{t-1}; \xi_n)$, where $N \coloneqq |\mathcal{N}(t)|$, and the last equality follows from the existence of extreme point solutions of a linear program. □

The condition in Proposition 2 is quite commonly satisfied by many application problems. For example, when all the feasibility sets $\mathcal{F}_t^{\text{rob}}$ and local cost functions f_t^{rob} are defined by convex functions, and the uncertainty vectors ξ_t only appear in the right-hand-sides of the constraints (e.g., [15]), then we can reformulate the MRCO into a DR-MCO in the form (1). Therefore, we use (2) as the standard formulation in the rest of the paper.

2.2 Approximation of Recursions

We now discuss the approximation of functions $\mathcal{Q}_t$ and Q_n in the recursions (2) and (3). Recall the relation $\mathcal{Q}_t(x_t) = \max_{p_t \in \mathcal{P}_t} \sum_{n \in \mathcal{N}(t+1)} p_{t,n} Q_n(x_t)$. The following lemma relates the Lipschitz continuity of the value functions Q_n and the cost-to-go function $\mathcal{Q}_t$.

Lemma 1. *For each stage $t < T$, if Q_n is l_n-Lipschitz continuous on $\mathcal{X}_t$ for each $n \in \mathcal{N}(t+1)$, then $\mathcal{Q}_t$ is L_t-Lipschitz continuous on $\mathcal{X}_t$ where $L_t = \max_{n \in \mathcal{N}(t+1)} l_n$.*

Proof. Take any two points $x_t^i \in \mathcal{X}_t$, $i = 1, 2$. Let $p_t^i \in \arg\max_{p_t \in \mathcal{P}_t} \sum_{n \in \mathcal{N}(t+1)} p_{t,n} Q_n(x^i)$ denote corresponding maximizers for $i = 1, 2$. We have $\mathcal{Q}_t(x_t^i) = \sum_{n \in \mathcal{N}(t+1)} p_{t,n}^i Q_n(x_t^i)$, where $p_{t,n}^i \geq 0$ and $\sum_{n \in \mathcal{N}(t+1)} p_{t,n}^i = 1$ for each $i = 1, 2$. Therefore,

$$\begin{aligned}\mathcal{Q}_t(x_t^1) - \mathcal{Q}_t(x_t^2) &= \sum_{n \in \mathcal{N}(t+1)} p_{t,n}^1 Q_n(x_t^1) - \sum_{n \in \mathcal{N}(t+1)} p_{t,n}^2 Q_n(x_t^2) \\ &\leq \sum_{n \in \mathcal{N}(t+1)} p_{n,t}^1 \left(Q_n(x_t^1) - Q_n(x_t^2) \right) \leq \sum_{n \in \mathcal{N}(t+1)} p_{n,t}^1 \cdot l_n \left\| x_t^1 - x_t^2 \right\| \leq L_t \left\| x_t^1 - x_t^2 \right\|,\end{aligned}$$

where $L_t = \max_{n \in \mathcal{N}(t+1)} l_n$. By exchanging the indices $i = 1, 2$, we similarly derive $\mathcal{Q}_t(x_t^2) - \mathcal{Q}_t(x_t^1) \leq \sum_{n \in \mathcal{N}(t+1)} p_{n,t}^2 \cdot l_n \left\| x_t^1 - x_t^2 \right\| \leq L_t \left\| x_t^1 - x_t^2 \right\|$, which completes the proof. □

Combining Lemma 1 and Proposition 1, we know that if the value functions are convex and Lipschitz continuous, then so are the cost-to-go functions. In such a case, we can use cutting plane method to build an under-approximation of the cost-to-go functions. To be precise, for each node $n \in \mathcal{N}(t+1)$, let $V_n(x_t)$ denote an affine function such that $Q_n(x_t) \geq V_n(x_t)$ for all $x_t \in \mathcal{X}_t$. Such affine function is referred to as a linear valid inequality or a linear cut for the value function, which is generated in the following way. Let $\underline{\mathcal{Q}}_{t+1}$ denote an under-approximation of the cost-to-go function $\mathcal{Q}_{t+1}$ and $\hat{x}_t \in \mathcal{X}_t$ a feasible state. For each $n \in \mathcal{N}(t+1)$, the Lagrangian dual problem

$$\begin{aligned} \sup_{\lambda_n} \min_{x_n, y_n, z_n} \quad & f_n(z_n, y_n, x_n) + \underline{\mathcal{Q}}_{t+1}(x_n) + \langle \lambda_n, \hat{x}_t - z_n \rangle \\ \text{s.t.} \quad & (x_n, y_n) \in \mathcal{F}_n \end{aligned} \tag{8}$$

gives an affine function $V_n(x_t) \coloneqq \underline{v}_n + \langle \hat{\lambda}_n, x_t - \hat{x}_t \rangle$, where $\hat{\lambda}_n$ is a dual solution of (8) and $\underline{v}_n \coloneqq \min\{f_n(z_n, y_n, x_n) + \underline{\mathcal{Q}}_{t+1}(x_n) + \langle \hat{\lambda}_n, \hat{x}_t - z_n \rangle : (x_n, y_n) \in \mathcal{F}_n\}$ is the associated value

to the problem (8). Then, by definition (3) and weak duality, we have for every $x_t \in \mathcal{X}_t$,

$$\begin{aligned} Q_n(x_t) &\geq \sup_{\lambda_n} \min_{x_n, y_n, z_n} \left\{ f_n(z_n, y_n, x_n) + \mathcal{Q}_{t+1}(x_n) + \langle \lambda_n, x_t - z_n \rangle : (x_n, y_n) \in \mathcal{F}_n \right\} \\ &\geq \sup_{\lambda_n} \min_{x_n, y_n, z_n} \left\{ f_n(z_n, y_n, x_n) + \underline{\mathcal{Q}}_{t+1}(x_n) + \langle \lambda_n, \hat{x}_t - z_n \rangle : (x_n, y_n) \in \mathcal{F}_n \right\} + \langle \lambda_n, x_t - \hat{x}_t \rangle \\ &\geq \min_{x_n, y_n, z_n} \left\{ f_n(z_n, y_n, x_n) + \underline{\mathcal{Q}}_{t+1}(x_n) + \langle \hat{\lambda}_n, x_t - z_n \rangle : (x_n, y_n) \in \mathcal{F}_n \right\} + \langle \hat{\lambda}_n, x_t - \hat{x}_t \rangle \\ &\geq \underline{v}_n + \langle \hat{\lambda}_n, x_t - \hat{x}_t \rangle. \end{aligned} \tag{9}$$

Therefore, $V_n(x_t)$ is a valid inequality for the value function $Q_n(x_t)$. The next proposition shows that we can combine linear cuts for value functions into a valid inequality for the cost-to-go function.

Proposition 3. *Fix a point $x_t^0 \in \mathcal{X}_t$ and a probability vector $p_t^0 \in \mathcal{P}_t$. If for each $n \in \mathcal{N}(t+1)$, $V_n(x_t) = \underline{v}_n + \langle \hat{\lambda}_n, x_t - \hat{x}_t \rangle$ is an l'_n-Lipschitz continuous valid inequality, i.e. $\|\hat{\lambda}_n\| \leq l'_n$, then $\mathcal{Q}_t(x_t) \geq \mathcal{V}_t(x_t) \coloneqq \sum_{n \in \mathcal{N}(t+1)} p^0_{t,n} V_n(x_t)$ for all $x_t \in \mathcal{X}_t$. Moreover, the valid inequality $\mathcal{V}_t$ is $L'_t \coloneqq (\sum_{n \in \mathcal{N}(t+1)} p^0_{t,n} l'_n)$-Lipschitz continuous such that for any $\delta > \gamma \coloneqq \mathcal{Q}_t(x_t^0) - \mathcal{V}_t(x_t^0)$, we have $\mathcal{Q}_t(x_t) \leq \mathcal{V}_t(x_t) + \delta$, for any point $x_t \in \mathcal{X}_t$ with $\|x_t - x_t^0\| \leq (\delta - \gamma)/(L_t + L'_t)$.*

Proof. The first claim follows from the definition

$$\mathcal{Q}_t(x_t) = \max_{p_t \in \mathcal{P}_t} \sum_{n \in \mathcal{N}(t+1)} p_{t,n} Q_n(x_t) \geq \sum_{n \in \mathcal{N}(t+1)} p^0_{t,n} Q_n(x_t) \geq \sum_{n \in \mathcal{N}(t+1)} p^0_{t,n} V_n(x_t) = \mathcal{V}_t(x_t).$$

The second claim follows from the Lipschitz continuity of the valid inequality and the cost-to-go function, i.e., for any point $x_t \in \mathcal{X}_t$ with $\|x_t - x_t^0\| \leq (\delta - \gamma)/(L_t + L'_t)$,

$$\mathcal{Q}_t(x_t) \leq \mathcal{Q}_t(x_t^0) + L_t \|x_t - x_t^0\| \leq \mathcal{V}_t(x_t) + (L_t + L'_t)\|x_t - x_t^0\| + \gamma \leq \mathcal{V}_t(x_t) + \delta,$$

where the first inequality is due to $\mathcal{Q}_t$ being L_t-Lipschitz continuous by Lemma 1. □

The proposition suggests that the combined linear cut is close to the cost-to-go function in a neighborhood, given that the gap $\mathcal{Q}_t(x_t^0) - \mathcal{V}_t(x_t^0)$ is small. However, the radius of such neighborhood depends on the Lipschitz constants of the linear cuts V_n, which are not necessarily bounded by the Lipschitz constants l_n of value functions Q_n, as is shown by the following example.

Example 1. *Consider a T-stage deterministic problem (i.e., $|\mathcal{N}(t)| = 1$ for all $t \in \mathcal{T}$) defined as*

$$\begin{aligned} Q_t(x_{t-1}) \coloneqq \min_{y_t, x_t} \quad & y_t + Q_{t+1}(x_t) \\ \text{s.t.} \quad & y_t \geq \max\{0, 1 - 2x_{t-1}\},\ x_t \leq x_{t-1} + \frac{1}{2},\ 0 \leq x_t \leq 1. \end{aligned}$$

Here, the convention $Q_{T+1}(x_t) \equiv 0$ is used to simplify the definition. Note that for each stage $t \in \mathcal{T}$, since $x_{t-1} \in [0,1]$, we have a feasible solution $x_t = 1/2$, which implies that $Q_t(x_{t-1}) \leq \min\{y_t : y_t \geq \max\{0, 1-2x_{t-1}\}$. By taking $y_t = \max\{0, 1-2x_{t-1}\}$ and using the fact that $Q_{t+1}(x_t) \geq 0$ recursively, we conclude that the cost-to-go functions are $Q_{t+1}(x_t) = \max\{0, 1-2x_t\}$ for all $t \in \mathcal{T}$. However, if we start our approximation with points $x_t^0 = 0$ for all stages $t \in \mathcal{T}$, then the linear cut $V_t(x_t) \coloneqq v_t + \langle \lambda_t, x_t - x_t^0 \rangle$ can be generated from the following dual problem at stage t:

$$\begin{aligned} v_t \coloneqq \max_{\lambda_t} \min_{z_t, y_t, x_t} \quad & y_t + \lambda_t(0 - z_t) + \underline{Q}_{t+1}^0(x_t) \\ \text{s.t.} \quad & y_t \geq \max\{0, 1-2z_t\},\ x_t \leq z_t + \frac{1}{2},\ 0 \leq x_t \leq 1. \end{aligned}$$

Thus the under-approximation of the value functions of stage T will be $\underline{Q}_T^0(x_{T-1}) = 1 - 2x_{T-1}$. Plugging in the under-approximation and the dual problem becomes

$$\begin{aligned} v_t = \max_{\lambda_t} \min_{z_t, y_t, x_t} \quad & y_t + \lambda_t(0 - z_t) + 1 - c_t x_t \\ \text{s.t.} \quad & y_t \geq \max\{0, 1-2z_t\},\ x_t \leq z_t + \frac{1}{2},\ 0 \leq x_t \leq 1, \end{aligned}$$

where $c_T = 2$. By solving the dual problem recursively, we conclude that $c_t = c_{t+1} + 2$ and hence $c_t = 2(T - t + 1)$. In other words, the under-approximation obtained after the first iteration has a Lipschitz constant greater than the actual one.

The example shows that, the Lipschitz constant of the generated linear cuts is not completely determined by the problem data - it also depends on how we approximate the value function during the algorithm by picking the state at which we generate the linear cut. This phenomenon of growing Lipschitz constant is undesirable for our complexity analysis in this paper, because naturally the complexity of an algorithm should be determined by the problem data, without the arbitrariness on how we solve the subproblem (or the output of a subproblem oracle). To overcome this problem, we consider a surrogate for the true problem (1) with the nice property of the bounded Lipschitz constant for value function approximations, which is referred to as the regularization of the original problem (1).

2.3 Regularization and Its Exactness

The potential gap between the actual Lipschitz constants of the value functions Q_n and the generated linear cuts V_n (Example 1) not only affects the complexity analysis, but may also negatively impact algorithmic performance. For this reason, we consider the regularization of value functions, in which an infimal convolution is conducted to bound the generated linear cuts. We begin with the recursive definition of regularized value functions and cost-to-go functions. Let $M_t > 0$ denote a regularization factor for each $t \in \mathcal{T} \setminus \{T\}$. The regularized cost-to-go function is defined as

$$\mathcal{Q}_t^{\mathrm{R}}(x_t) \coloneqq \max_{p_t \in \mathcal{P}_t} \sum_{n \in \mathcal{N}(t+1)} p_{t,n} Q_n^{\mathrm{R}}(x_t), \tag{10}$$

where the regularized value function Q_n^{R} for each subproblem $n \in \mathcal{N}(t+1)$ is defined as

$$Q_n^{\mathrm{R}}(x_t) \coloneqq \min_{x_n, y_n, z_n} \quad f_n(z_n, y_n, x_n) + \mathcal{Q}_{t+1}^{\mathrm{R}}(x_n) + M_t \|x_t - z_n\| \tag{11}$$
$$\text{s.t.} \quad (x_n, y_n) \in \mathcal{F}_n.$$

We set the same convention that $\mathcal{Q}_T^{\mathrm{R}}(x_T) \equiv 0$. Note that for nodes in the last stage $n \in \mathcal{N}(T)$ and any $x_{T-1} \in \mathcal{X}_{T-1}$, since $Q_n^{\mathrm{R}}(x_{T-1}) = \min_{z_n} \left\{ Q_n(z_n) + M_{T-1} \|x_{T-1} - z_n\| \right\} \leq Q_n(x_{T-1})$ is an infimal convolution of Q_n and the norm function $M_{T-1} \|\cdot\|$, hence it is M_t-Lipschitz continuous. The next proposition shows that the functions Q_n^{R} and $\mathcal{Q}_t^{\mathrm{R}}$ are Lipschitz continuous envelopes of Q_n and $\mathcal{Q}_t$, respectively, for all $n \in \mathcal{N}(t)$ and $t \in \mathcal{T}$.

Proposition 4. *For every node $n \in \mathcal{N}(t+1)$, $t \in \mathcal{T} \setminus \{T\}$, the regularized value function Q_n^{R} is M_t-Lipschitz continuous, and $Q_n^{\mathrm{R}}(x_t) \leq Q_n(x_t)$, $\forall\, x_t \in \mathcal{X}_t$. Moreover, if Q_n is M_t'-Lipschitz continuous with $M_t' < M_t$ for all $n \in \mathcal{N}(t+1)$ and $t < T$, then $Q_n(x_t) = Q_n^{\mathrm{R}}(x_t)$, $\forall\, x_t \in \mathcal{X}_t$.*

Proof. For any node $n \in \mathcal{N}(t+1)$ for some $t \in \mathcal{T}$, pick any $x_t^1, x_t^2 \in \mathcal{X}_t$. Let x_n^i, y_n^i, z_n^i denote the solutions in the definition (10) associated with x_t^i for $i = 1$ and 2, respectively. Then,

$$\begin{aligned} Q_n^{\mathrm{R}}(x_t^1) - Q_n^{\mathrm{R}}(x_t^2) &= f_n(z_n^1, y_n^1, x_n^1) + \mathcal{Q}_{t+1}^{\mathrm{R}}(x_n^1) + M_t\|x_t^1 - z_n^1\| - f_n(z_n^2, y_n^2, x_n^2) - \mathcal{Q}_{t+1}^{\mathrm{R}}(x_n^2) - M_t\|x_t^2 - z_n^2\| \\ &\leq f_n(z_n^2, y_n^2, x_n^2) + \mathcal{Q}_{t+1}^{\mathrm{R}}(x_n^2) + M_t\|x_t^1 - z_n^2\| - f_n(z_n^2, y_n^2, x_n^2) - \mathcal{Q}_{t+1}^{\mathrm{R}}(x_n^2) - M_t\|x_t^2 - z_n^2\| \\ &= M_t \left(\|x_t^1 - z_n^2\| - \|x_t^2 - z_n^2\| \right) \leq M_t \|x_t^1 - x_t^2\|. \end{aligned}$$

Similarly by exchanging the indices $i = 1, 2$, we derive that $Q_n^{\mathrm{R}}(x_t^2) - Q_n^{\mathrm{R}}(x_t^1) \leq M_t \left\|x_t^1 - x_t^2\right\|$, which shows that Q_n^{R} is M_t-Lipschitz continuous.

We next prove the inequalities $Q_n^{\mathrm{R}}(x_t) \leq Q_n(x_t)$, $x_t \in \mathcal{X}_t$ recursively. For nodes in the last stage $n \in \mathcal{N}(T)$, we already show that $Q_n^{\mathrm{R}}(x_{T-1}) \leq Q_n(x_{T-1})$ for any $x_{T-1} \in \mathcal{X}_{T-1}$. By definition (2), we have $\mathcal{Q}_{T-1}^{\mathrm{R}}(x_{T-1}) \leq \mathcal{Q}_{T-1}(x_{T-1})$. Now suppose $\mathcal{Q}_t^{\mathrm{R}}(x_t) \leq \mathcal{Q}_t(x_t)$ for some $t \in \mathcal{T} \setminus \{T\}$. Then,

$$\begin{aligned} Q_n^{\mathrm{R}}(x_t) &= \min_{x_n, y_n, z_n} \left\{ f_n(z_n, y_n, x_n) + \mathcal{Q}_{t+1}^{\mathrm{R}}(x_n) + M_t \|x_t - z_n\| : (x_n, y_n) \in \mathcal{F}_n \right\} \\ &\leq \min_{x_n, y_n, z_n} \left\{ f_n(z_n, y_n, x_n) + \mathcal{Q}_{t+1}(x_n) + M_t \|x_t - z_n\| : (x_n, y_n) \in \mathcal{F}_n \right\} \\ &\leq \min_{x_n, y_n} \left\{ f_n(x_t, y_n, x_n) + \mathcal{Q}_{t+1}(x_n) : (x_n, y_n) \in \mathcal{F}_n \right\} = Q_n(x_t), \end{aligned} \tag{12}$$

where the last inequality is due to the fact that $z_n = x_t$ is a feasible solution to the minimization problem. Now by definitions (2) and (10), we have $\mathcal{Q}_t^{\mathrm{R}}(x_t) \leq \mathcal{Q}_t(x_t)$ as well. We have thus shown recursively that $Q_n^{\mathrm{R}}(x_t) \leq Q_n(x_t)$ for any $x_t \in \mathcal{X}_t$.

For the last statement, we claim that if $\mathcal{Q}_{t+1}(x_{t+1}) = \mathcal{Q}_{t+1}^{\mathrm{R}}(x_{t+1})$ for any $x_{t+1} \in \mathcal{X}_{t+1}$ and $Q_n(x_t)$ is M_t'-Lipschitz continuous, then $Q_n^{\mathrm{R}}(x_t) = Q_n(x_t)$ for any $x_t \in \mathcal{X}_t$. To see this claim, note that both inequalities in the above argument (12) become equalities: the first

one follows from the assumption, and the second one is due to that $z_n = x_t$ is the unique solution to the minimization problem. Therefore, we can apply the claim recursively as well to see that $Q_n^{\mathrm{R}}(x_t) = Q_n(x_t)$ for all $x_t \in \mathcal{X}_t$, $n \in \mathcal{N}(t+1)$, and all $t \in \mathcal{T} \setminus \{T\}$. □

The proposition implies that if we already have an estimate of the Lipschitz constants of value functions for the original problem (3), then the regularization problem (10) is equivalent in the sense that all the regularized value functions are equal to the original ones. Thus solving the regularization (10) does not compromise any feasibility or optimality in this case. An important difference that regularization brings is that all the linear cuts generated for the regularized value functions are all M_t-Lipschitz continuous, as shown by the following proposition.

Proposition 5. *Let $\underline{\mathcal{Q}}_{t+1}$ denote any under-approximation of the cost-to-go function $\mathcal{Q}_{t+1}^{\mathrm{R}}$ and $\hat{x}_t \in \mathcal{X}_t$ a feasible state. The linear cut $V_n(x_t) \coloneqq \underline{v}_n + \left\langle \hat{\lambda}_n, x_t - \hat{x}_t \right\rangle$ is M_t-Lipschitz continuous, where $\hat{\lambda}_n$ is a dual solution and $\underline{v}_n$ is its associated value to the Lagrangian dual problem*

$$\begin{aligned} \sup_{\lambda_n} \min_{x_n, y_n, z_n, w_n} \quad & f_n(z_n, y_n, x_n) + \underline{\mathcal{Q}}_{t+1}^{\mathrm{R}}(x_n) + M_t \left\| \hat{x}_t - w_n \right\| + \langle \lambda_n, w_n - z_n \rangle \\ \text{s.t.} \quad & (x_n, y_n) \in \mathcal{F}_n. \end{aligned} \tag{13}$$

Proof. Let $\hat{\lambda}_n$ denote a feasible solution, i.e., the associated value $\underline{v}_n > -\infty$. Then using the optimality condition for the variable w_n, we have $-\hat{\lambda}_n \in \partial_{w_n}(M_t \left\| \hat{x}_t - w_n \right\|)$, where all the norms of elements in the subdifferential set are bounded by $M_t > 0$. Therefore we must have $\|\hat{\lambda}_n\|_* \leq M_t$ and $V_n(x_t)$ is M_t-Lipschitz continuous. □

We remark that after regularization, the Lipschitz constant of the linear cut can always be bounded by the regularization factor M_t. Together with Proposition 4, this implies that even for those problems with Lipschitz continuous value functions, regularization may help with the approximation by avoiding linear cuts that have larger Lipschitz constant than that of the value function (cf. Example 1).

2.3.1 Non-Lipschitz-Continuous Value Function Cases

We spend the rest of this section showing that the regularization can be exact for a broader class of problems than those that already have Lipschitz continuous value functions, which extends our analysis to some problems even without relatively complete recourse assumptions. It is known that $\mathcal{P}_n$ can be assumed to be convex without affecting the value functions. Without loss of generality, we write $\mathcal{P}_t = \mathcal{K}_t \cap \Delta^{|\mathcal{N}(t+1)|}$ where $\mathcal{K}_t$ is a closed convex cone contained in the nonnegative orthant. Then the value function for node $n \in \mathcal{N}(t)$ can be

written as

$$
\begin{aligned}
Q_n(x_{t-1}) = \min_{x_n, y_n, c_n} \; & f_n(x_t, y_n, x_n) + c_n, \\
\text{s.t.} \; & \big(c_n - Q_m(x_n)\big)_{m \in \mathcal{N}(t)} \in \mathcal{K}_t^*, \\
& (x_n, y_n) \in \mathcal{F}_n,
\end{aligned}
\tag{14}
$$

by taking the dual of the maximization over transition probability vectors $p_t \in \mathcal{P}_t$ in the definition (2). To get an extensive formulation, we define $\tilde{\mathcal{N}}(t) \coloneqq \prod_{t'=1}^{t} \mathcal{N}(t')$ as an extended set of nodes, such that each node $n \in \tilde{\mathcal{N}}(t)$ is determined by a vector $(n_1, \ldots, n_t)$ where $n_{t'} \in \mathcal{N}(t')$ for each $t' \leq t$. The extended set $\tilde{\mathcal{N}} \coloneqq \cup_{t=1}^{T} \tilde{\mathcal{N}}(t)$ naturally has a tree structure, so we use $a(n)$, $\mathcal{C}(n)$, and $\mathcal{D}(n)$ to denote the parent node, the set of child nodes, and the set of all descendent nodes of a node $n \in \tilde{\mathcal{N}}$, respectively. For notational convenience, we still use $n = 1$ to denote the root node, corresponding to the deterministic first stage, and $t(n)$ to denote the associated stage to a node $n \in \tilde{\mathcal{N}}$. Now by substituting the formulation (14) into the recursion (2) recursively, we obtain an extensive formulation of the problem (1).

$$
\begin{aligned}
v^{\text{ext}} \coloneqq \min \quad & f_1(x_0, y_1, x_1) + c_1 & & \\
\text{s.t.} \quad & (x_n, y_n) \in \mathcal{F}_n, & & \forall\, n \in \tilde{\mathcal{N}}, \\
& (c_n - q_{nm})_{m \in \mathcal{C}(n)} \in \mathcal{K}_{t(n)}^*, & & \forall\, n \in \tilde{\mathcal{N}}, \\
& q_{nm} \geq f_m(x_n, y_m, x_m) + c_m, & & \forall\, n = a(m),\ m \in \tilde{\mathcal{N}}.
\end{aligned}
\tag{15}
$$

We can develop exact penalization on the extensive formulation now. Let $\sigma > 0$ denote a penalty factor for non-root nodes $m \neq 1$. Then the penalization value is defined by

$$
\begin{aligned}
v^{\text{pen}} \coloneqq \min \quad & f_1(x_0, y_1, x_1) + c_1 + \sum_{m \neq 1} \sigma \left\| x_{a(m)} - z_m \right\| & & \\
\text{s.t.} \quad & (x_n, y_n) \in \mathcal{F}_n, & & \forall\, n \in \tilde{\mathcal{N}}, \\
& (c_n - q_{nm})_{m \in \mathcal{C}(n)} \in \mathcal{K}_{t(n)}^*, & & \forall\, n \in \tilde{\mathcal{N}}, \\
& q_{nm} \geq f_m(z_m, y_m, x_m) + c_m, & & \forall\, n = a(m),\ m \in \tilde{\mathcal{N}}.
\end{aligned}
\tag{16}
$$

The penalization objective value v^{pen} depends on the choice of the penalty factor σ. We make the following assumption on the exactness of this penalization.

Assumption 1. *There exists a penalty factor $\sigma > 0$ such that $v^{\text{pen}} = v^{\text{ext}}$. Moreover, any optimal solution to the penalization (16) satisfies $z_m = x_{a(m)}$ for all $m \neq 1 \in \tilde{\mathcal{N}}$.*

We remark that this assumption is satisfied given some constraint qualification, e.g., Slater condition for (15) assuming convexity, or if all the local cost functions f_n, the feasibility sets $\mathcal{F}_n$, and the cones $\mathcal{K}_t$ are polyhedral. By the introduction of cost-to-go functions

$\mathcal{Q}_t$, we can define the recursive formulation as follows.

$$v^{\text{pen}} = \min_{(x_1,y_1)\in\mathcal{F}_1} \quad f_1(x_{a(1)}, y_1, x_1) + \mathcal{Q}_1^{\text{pen}}(x_1), \tag{17}$$

$$\begin{aligned}
\mathcal{Q}_1^{\text{pen}}(x_1) \coloneqq \min \quad & c_1 + \sum_{m\neq 1} \sigma \left\| x_{a(m)} - z_m \right\| && (18)\\
\text{s.t.} \quad & (x_m, y_m) \in \mathcal{F}_m, && \forall\, m \neq 1 \in \tilde{\mathcal{N}},\\
& (c_m - q_{ml})_{l\in\mathcal{C}(m)} \in \mathcal{K}^*_{t(m)}, && \forall\, m \in \tilde{\mathcal{N}},\\
& q_{ml} \geq f_l(z_l, y_l, x_l) + c_l, && \forall\, m = a(l),\ l \in \tilde{\mathcal{N}},\\
= \min_{z_m} & \sum_{m\neq 1} \sigma \left\| x_{a(m)} - z_m \right\| + \min_{c_1, q_{1m}} \min_{\substack{c_m, q_{ml},\\ x_m, y_m}} \quad c_1 && (19)\\
\text{s.t.} \quad & (x_m, y_m) \in \mathcal{F}_m, && \forall\, m \neq 1 \in \tilde{\mathcal{N}},\\
& (c_m - q_{ml})_{l\in\mathcal{C}(m)} \in \mathcal{K}^*_{t(m)}, && \forall\, m \in \tilde{\mathcal{N}},\\
& q_{ml} \geq f_l(z_l, y_l, x_l) + c_l, && \forall\, m = a(l),\ l \in \tilde{\mathcal{N}}.
\end{aligned}$$

Note that by the definition of the dual cone $\mathcal{K}_1^*$, the cost-to-go function can be rewritten as

$$\mathcal{Q}_r^{\text{pen}}(x_r) = \min_{z_m} \sum_{m\neq r} \sigma \left\| x_{a(m)} - z_m \right\| + \max_{p_r\in\mathcal{P}_r} \min_{x_m, y_m} \quad \sum_{m\in\mathcal{C}(r)} p_{rm}(f_m(z_m, y_m, x_m) + c_m) \tag{20}$$

$$\begin{aligned}
\text{s.t.} \quad & (x_m, y_m) \in \mathcal{F}_m, && \forall\, m \neq 1 \in \tilde{\mathcal{N}},\\
& (c_m - q_{ml})_{l\in\mathcal{C}(m)} \in \mathcal{K}^*_{t(m)}, && \forall\, m \neq 1 \in \tilde{\mathcal{N}},\\
& q_{ml} \geq f_l(z_l, y_l, x_l) + c_l, && \forall\, m = a(l) \neq 1,\ l \in \tilde{\mathcal{N}}.\\
= \max_{p_r\in\mathcal{P}_r} \min \quad & \sum_{m\in\mathcal{C}(r)} p_{rm}(f_m(z_m, y_m, x_m) + c_m) + \sum_{m\neq r} \sigma \left\| x_{a(m)} - z_m \right\| && (21)\\
\text{s.t.} \quad & (x_m, y_m) \in \mathcal{F}_m, && \forall\, m \neq 1 \in \tilde{\mathcal{N}},\\
& (c_m - q_{ml})_{l\in\mathcal{C}(m)} \in \mathcal{K}^*_{t(m)}, && \forall\, m \neq 1 \in \tilde{\mathcal{N}},\\
& q_{ml} \geq f_l(z_l, y_l, x_l) + c_l, && \forall\, m = a(l) \neq 1,\ l \in \tilde{\mathcal{N}}.
\end{aligned}$$

The last step of exchanging min and max is due to the convexity of the problem and the

compactness of the uncertainty set $\mathcal{P}_1$. Now, we define the cost-to-go functions for $m \in \mathcal{C}(1)$:

$$\mathcal{Q}_1^{\text{pen}}(x_1) = \max_{p_1 \in \mathcal{P}_1} \min_{x_m, y_m, z_m} \sum_{m \in \mathcal{C}(1)} \left[p_{1m}(f_m(z_m, y_m, x_m) + \mathcal{Q}_m^{\text{pen}}(x_m)) + \sigma \left\| x_1 - z_m \right\| \right], \quad (22)$$

$$\begin{aligned} \mathcal{Q}_m^{\text{pen}}(x_m) \coloneqq \min \quad & c_m + \sum_{l \in \mathcal{D}(m)} \sigma \left\| x_{a(l)} - z_l \right\| && (23) \\ \text{s.t.} \quad & (x_l, y_l) \in \mathcal{F}_l, && \forall\, l \in \mathcal{D}(m), \\ & (c_l - q_{lk})_{k \in \mathcal{C}(l)} \in \mathcal{K}_l^*, && \forall\, l \in \mathcal{D}(m), \\ & q_{lk} \geq f_k(z_k, y_k, x_k) + c_k, && \forall\, l = a(k) \in \mathcal{D}(m). \end{aligned}$$

By repeating the above definition of cost-to-go functions, we can have the recursive formulation of the penalization

$$v^{\text{pen}} = \min_{(x_1, y_1) \in \mathcal{F}_1} \quad f_1(x_0, y_1, x_1) + \mathcal{Q}_1^{\text{pen}}(x_1), \quad (24)$$

$$\begin{aligned} \mathcal{Q}_n^{\text{pen}}(x_n) = \max_{p_n \in \mathcal{P}_n} \min \quad & \sum_{m \in \mathcal{C}(n)} \left[p_{nm}(f_m(z_m, y_m, x_m) + \mathcal{Q}_m^{\text{pen}}(x_m)) + \sigma \left\| x_n - z_m \right\| \right], && (25) \\ \text{s.t.} \quad & (x_m, y_m) \in \mathcal{F}_m, \ \forall\, m \in \mathcal{C}(n). \end{aligned}$$

Note by this definition, $\mathcal{Q}_n \equiv 0$ for all leaf nodes $n \in \tilde{\mathcal{N}}$ with $\mathcal{C}(n) = \varnothing$.

While the penalization is known to be exact under Assumption 1, the evaluation of the cost-to-go function at a given point may be challenging. In the sequel, we show that we can replace the penalization formulation with regularization under the following assumption:

Assumption 2. *There exists a constant $0 < c \leq 1$ such that for any $n \in \tilde{\mathcal{N}}$ and $x_n \in \mathcal{X}_n$, there exists a maximizer $\hat{p}_t \in \mathcal{P}_{t(n)}$ in (25) with each component satisfying either $\hat{p}_{nm} = 0$ or $\hat{p}_{nm} \geq c$, $m \in \mathcal{C}(n)$.*

Let $\tau > 0$ denote a regularization factor for all non-root nodes. We now define the regularization cost-to-go functions recursively as

$$v^{\text{reg}} = \min_{(x_1, y_1) \in \mathcal{F}_1} \quad f_1(x_0, y_1, x_1) + \mathcal{Q}_1^{\text{reg}}(x_1), \quad (26)$$

$$\begin{aligned} \mathcal{Q}_n^{\text{reg}}(x_n) = \max_{p_n \in \mathcal{P}_n} \min \quad & \sum_{m \in \mathcal{C}(n)} p_{nm} \left[f_m(z_m, y_m, x_m) + \mathcal{Q}_m^{\text{reg}}(x_m) + \tau \left\| x_n - z_m \right\| \right], && (27) \\ \text{s.t.} \quad & (x_m, y_m) \in \mathcal{F}_m, \ n \in \tilde{\mathcal{N}}, m \in \mathcal{C}(n). \end{aligned}$$

Proposition 6. *Given the constant $c > 0$ stated in Assumption 2, if $\tau \geq \sigma / c$, then the regularization is exact, i.e., $v^{\text{reg}} = v^{\text{pen}} = v^{\text{ext}}$ and the corresponding set of first stage minimizers are the same.*

Proof. For ease of notation, we denote

$$\hat{\mathcal{F}}_1 \coloneqq \operatorname*{arg\,min}_{(x_1, y_1) \in \mathcal{F}_1} \{ f_1(x_0, y_1, x_1) + \mathcal{Q}_1(x_1) \},$$

$$\hat{\mathcal{F}}_1^{\text{pen}} \coloneqq \underset{(x_1,y_1)\in\mathcal{F}_1}{\arg\min} \{f_1(x_0,y_1,x_1) + \mathcal{Q}_1^{\text{pen}}(x_1)\},$$

and

$$\hat{\mathcal{F}}_1^{\text{reg}} \coloneqq \underset{(x_1,y_1)\in\mathcal{F}_1}{\arg\min} \{f_1(x_0,y_1,x_1) + \mathcal{Q}_1^{\text{reg}}(x_1)\},$$

respectively. By definition, we have $\mathcal{Q}_n^{\text{reg}}(x_n) \leq \mathcal{Q}_n(x_n)$ for all $n \in \tilde{\mathcal{N}}$ and $x_n \in \mathcal{X}_n$, which implies $v^{\text{reg}} \leq v^{\text{ext}}$. Assumption 1 of exact penalization implies that $v^{\text{pen}} = v^{\text{ext}}$ and $\hat{\mathcal{F}}_1^{\text{pen}} = \hat{\mathcal{F}}_1$. We claim that $\mathcal{Q}_n^{\text{reg}}(x_n) \geq \mathcal{Q}_n^{\text{pen}}(x_n)$ for all nodes $n \in \tilde{\mathcal{N}}$ and states $x_n \in \mathcal{X}_n$. Given this claim, it follows that $v^{\text{reg}} \geq v^{\text{pen}}$ and hence $v^{\text{reg}} = v^{\text{pen}} = v^{\text{ext}}$. We thus conclude $\hat{\mathcal{F}}_1 \subseteq \hat{\mathcal{F}}_1^{\text{reg}} \subseteq \hat{\mathcal{F}}_1^{\text{pen}}$, implying the equality of these three sets.

We prove the claim recursively. For any leaf node n, $\mathcal{Q}_n^{\text{reg}}(x_n) = \mathcal{Q}_n^{\text{pen}}(x_n) = 0$ since $\mathcal{C}(n) = \varnothing$. Now assume that the claim $\mathcal{Q}_m^{\text{reg}}(x_m) \geq \mathcal{Q}_m^{\text{pen}}(x_m)$ holds for all descendent nodes $m \in \mathcal{D}(n)$. Let $p_n \in \mathcal{P}_{t(n)}$ denote a maximizer associated with the state $x_n \in \mathcal{X}_n$ in the definition (25). Thus

$$\begin{aligned}\mathcal{Q}_n^{\text{pen}}(x_n) = \sum_{\substack{m\in\mathcal{C}(n):\\ p_{nm}\neq 0}} \min \quad & \sum_{m\in\mathcal{C}(n)} p_{nm}\left(f_m(z_m,y_m,x_m) + \mathcal{Q}_m^{\text{pen}}(x_m)\right) + \sigma\left\|x_n - z_m\right\|, \\ \text{s.t.} \quad & (x_m,y_m)\in\mathcal{F}_m,\ \forall\, m\in\mathcal{C}(n).\end{aligned}$$

By Assumption 2, $p_{nm} \geq c$. This implies that

$$\begin{aligned}\mathcal{Q}_n^{\text{pen}}(x_n) &\leq \sum_{\substack{m\in\mathcal{C}(n):\\ p_{nm}\neq 0}} p_{nm}\cdot\min\left\{f_m(z_m,y_m,x_m) + \mathcal{Q}_m^{\text{pen}}(x_m) + \tau\left\|x_n - z_m\right\| : (x_m,y_m)\in\mathcal{F}_m, m\in\mathcal{C}(n)\right\} \\ &\leq \sum_{\substack{m\in\mathcal{C}(n):\\ p_{nm}\neq 0}} p_{nm}\cdot\min\left\{f_m(z_m,y_m,x_m) + \mathcal{Q}_m^{\text{reg}}(x_m) + \tau\left\|x_n - z_m\right\| : (x_m,y_m)\in\mathcal{F}_m, m\in\mathcal{C}(n)\right\} \\ &\leq \max_{p_n\in\mathcal{P}_n} \min\left\{\sum_{m\in\mathcal{C}(n)} p_{nm}\left[f_m(z_m,y_m,x_m) + \mathcal{Q}_m^{\text{reg}}(x_m) + \tau\left\|x_n - z_m\right\|\right] : (x_m,y_m)\in\mathcal{F}_m, m\in\mathcal{C}(n)\right\} \\ &= \mathcal{Q}_n^{\text{reg}}(x_n).\end{aligned}$$

The first inequality is due to $p_{nm}\tau \geq c\tau \geq \sigma$; the second inequality is due to the recursion hypothesis; the third inequality is due to the definition of maximum in the worst-case probability distribution. Thus we have shown $\mathcal{Q}_n^{\text{reg}}(x_n) \geq \mathcal{Q}_n^{\text{pen}}(x_n)$, which finishes the proof through recursion. □

We remark that although in general Assumption 2 is not easy to verify, it holds in some common cases where the uncertainty set does not contain any point with zero component, or the case where all the uncertainty sets and the subproblems are polyhedral (e.g., RDDP in [15]).

3 Algorithms and Complexity Analysis

In this section, we first define single stage subproblem oracles, based on which we define the notion of complexity of the algorithms. We describe two versions of our proposed dual dynamic programming algorithm with sequential and nonsequential stage selection strategies, respectively. The complexity upper bound for each algorithm is then presented, and finally we provide a complexity lower bound for both algorithms, which shows the upper complexity bounds are essentially tight.

3.1 Single Stage Subproblem Oracles

A subproblem oracle is an oracle that gives a solution to the subproblem given its own information as well as the data generated by the algorithm. The single stage subproblem oracles (SSSO) used in this paper solve an approximation of the problem given by (10) and (11) (or given by (2) and (3) if $M_t = +\infty$, i.e., no regularization is conducted) for some stage $t \in \mathcal{T}$.

Definition 1 (Initial stage subproblem oracle)**.** *Let $\underline{\mathcal{Q}}_1, \overline{\mathcal{Q}}_1 : \mathcal{X}_1 \to \bar{\mathbb{R}}$ denote two closed convex functions, representing an under-approximation and an over-approximation of the expected cost-to-go function $\mathcal{Q}_1^{\mathrm{R}}$ in (10), respectively. Consider the following subproblem for the first stage $t = 1$,*

$$\min_{(x_1,y_1)\in\mathcal{F}_1} f_1(x_0, y_1, x_1) + \underline{\mathcal{Q}}_1(x_1), \tag{I}$$

where x_0 is a given parameter. The initial stage subproblem oracle provides an optimal solution (x_1, y_1) to (I) and calculates the approximation gap $\gamma_1 := \overline{\mathcal{Q}}_1(x_1) - \underline{\mathcal{Q}}_1(x_1)$ at the solution. We thus define the subproblem oracle formally as the map $\mathscr{O}_1 : (\underline{\mathcal{Q}}_1, \overline{\mathcal{Q}}_1) \mapsto (x_1, y_1; \gamma_1)$.

Definition 2 (Noninitial stage subproblem oracle)**.** *Let $\underline{\mathcal{Q}}_t, \overline{\mathcal{Q}}_t : \mathcal{X}_t \to \bar{\mathbb{R}}$ denote two closed convex functions, representing an under-approximation and an over-approximation of the expected cost-to-go function $\mathcal{Q}_t^{\mathrm{R}}$ in (10), respectively, for some stage $t > 1$. Then given a feasible state $x_{t-1} \in \mathcal{X}_{t-1}$, consider the following subproblems associated with the stage t with $\Theta_t = \underline{\mathcal{Q}}_t$ or $\Theta_t = \overline{\mathcal{Q}}_t$:*

$$\max_{p_{t-1}\in\mathcal{P}_{t-1}} \sum_{n\in\mathcal{N}(t)} p_{t-1,n} \left(\min_{\substack{(x_n,y_n)\in\mathcal{F}_n,\\ z_n\in\mathbb{R}^{d_{t-1}}}} f_n(z_n, y_n, x_n) + \Theta_t(x_n) + M_{t-1}\|x_{t-1} - z_n\| \right). \tag{N}$$

The noninitial stage subproblem oracle provides a feasible state $x_t \in \mathcal{X}_t$, an M_{t-1}-Lipschitz continuous linear cut $\mathcal{V}_{t-1}(\cdot)$, and an over-estimate value v_{t-1} such that

- *they are valid, i.e., $\mathcal{V}_{t-1}(x) \le \mathcal{Q}_{t-1}^{\mathrm{R}}(x)$ for any $x \in \mathcal{X}_{t-1}$ and $v_{t-1} \ge \mathcal{Q}_{t-1}^{\mathrm{R}}(x_{t-1})$;*
- *the gap is controlled, i.e., $v_{t-1} - \mathcal{V}_{t-1}(x_{t-1}) \le \gamma_t := \overline{\mathcal{Q}}_t(x_t) - \underline{\mathcal{Q}}_t(x_t)$.*

We thus define the subproblem oracle formally as the map $\mathscr{O}_t : (x_{t-1}, \underline{\mathcal{Q}}_t, \overline{\mathcal{Q}}_t) \mapsto (\mathcal{V}_{t-1}, v_{t-1}, x_t; \gamma_t)$.

The initial stage subproblem oracle in Definition 1 represents the solution procedure of the first stage problem (I). In contrast, Definition 2 may be less intuitive. For this reason, we propose a possible realization of the noninitial stage subproblem oracle following the discussion in Section 2.2.

We first consider the under-approximation in the subproblem oracle $\Theta_t = \underline{\mathcal{Q}}_t$. Recall that the Lagrangian dual for each inner minimization problem in (N) gives a primal-dual solution pair $(\hat{x}_n, \hat{y}_n, \hat{z}_n; \hat{\lambda}_n)$ and its associated value $\underline{v}_n$, as in (13). The linear cut $V_n(x) = \underline{v}_n + \left\langle \hat{\lambda}_n, x - x_{t-1} \right\rangle$ is valid for the value function Q_n^{R} and by Proposition 5, it is M_{t-1}-Lipschitz continuous. Now we define a linear cut $\mathcal{V}_{t-1}(x) \coloneqq \sum_{n\in\mathcal{N}(t)} \hat{p}_{t-1,n} V_n(x)$ where $\hat{p}_{t-1} \in \arg\max_{p_{t-1}\in\mathcal{P}_{t-1}} \sum_{n\in\mathcal{N}(t)} p_{t-1,n} V_n(x_{t-1})$ is a probability vector maximizer at the current state x_{t-1}. Then by Proposition 3, $\mathcal{V}_{t-1}$ is a valid linear cut for $\mathcal{Q}_{t-1}^{\mathrm{R}}$ with a Lipschitz constant M_{t-1}.

For the over-estimate value v_{t-1}, we consider the over-approximation in the subproblem oracle $\Theta_t = \overline{\mathcal{Q}}_t$. Since by assumption $\overline{\mathcal{Q}}_t(x) \geq \mathcal{Q}_t^{\mathrm{R}}(x)$ for all $x \in \mathcal{X}_t$, the optimal value v_{t-1} of (N) satisfies $v_{t-1} \geq \mathcal{Q}_{t-1}^{\mathrm{R}}(x_{t-1})$ by definition. Moreover, suppose the gap at each primal solution $\hat{x}_n$ is $\gamma_n \coloneqq \overline{\mathcal{Q}}_t(x_n) - \underline{\mathcal{Q}}_t(x_n)$. We pick the node index n^* such that the state $\hat{x}_{n^*}$ has the largest approximation gap γ_{n^*}, and set $x_t = \hat{x}_{n^*}$, $\gamma_t = \gamma_{n^*}$. Consequently, we have

$$
\begin{aligned}
v_{t-1} &= \max_{p_{t-1}\in\mathcal{P}_{t-1}} \sum_{n\in\mathcal{N}(t)} p_{t-1,n} \left(\min_{\substack{(x_n,y_n)\in\mathcal{F}_n,\\ z_n\in\mathbb{R}^{d_{t-1}}}} f_n(z_n, y_n, x_n) + \overline{\mathcal{Q}}_t(x_n) + M_{t-1} \left\| x_{t-1} - z_n \right\| \right) \\
&\leq \max_{p_{t-1}\in\mathcal{P}_{t-1}} \sum_{n\in\mathcal{N}(t)} p_{t-1,n} \left(f_n(\hat{z}_n, \hat{y}_n, \hat{x}_n) + \overline{\mathcal{Q}}_t(\hat{x}_n) + M_{t-1} \left\| x_{t-1} - \hat{z}_n \right\| \right) \\
&= \max_{p_{t-1}\in\mathcal{P}_{t-1}} \sum_{n\in\mathcal{N}(t)} p_{t-1,n} \left(f_n(\hat{z}_n, \hat{y}_n, \hat{x}_n) + \underline{\mathcal{Q}}_t(\hat{x}_n) + \gamma_n + M_{t-1} \left\| x_{t-1} - \hat{z}_n \right\| \right) \\
&\leq \gamma_t + \max_{p_{t-1}\in\mathcal{P}_{t-1}} \sum_{n\in\mathcal{N}(t)} p_{t-1,n} \left(f_n(\hat{z}_n, \hat{y}_n, \hat{x}_n) + \underline{\mathcal{Q}}_t(\hat{x}_n) + M_{t-1} \left\| x_{t-1} - \hat{z}_n \right\| \right) = \gamma_t + \mathcal{V}_{t-1}(x_{t-1}).
\end{aligned}
$$

Therefore, the condition $v_{t-1} - \mathcal{V}_{t-1}(x_{t-1}) \leq \gamma_t$ is satisfied. We summarize the above realization of the noninitial stage subproblem oracles in Algorithm 1.

We remark that Algorithm 1 is not the only way to realize SSSO. For example, it is discussed in [15] that the single stage subproblem of MRCO (5) can sometimes be reformulated as mixed-integer linear program, which may then be solved by branch-and-bound type algorithms. Therefore, the above definition of SSSO avoids restriction of the method used to solve (N). Besides, with SSSO, the complexity analysis will better reflect the computation time as the for-loop in Algorithm 1 can be easily parallelized. We also show in the next section that SSSO enables us to introduce a nonsequential dual dynamic programming algorithm.

Algorithm 1 A Realization of Noninitial Stage Subproblem Oracle

Require: State $x_{t-1} \in \mathcal{X}_{t-1}$, approximations $\underline{\mathcal{Q}}_t, \overline{\mathcal{Q}}_t : \underline{\mathcal{Q}}_t(x) \le \mathcal{Q}_t^{\mathrm{R}}(x) \le \overline{\mathcal{Q}}_t(x),\ \forall x \in \mathcal{X}_t$, $t > 1$

Ensure: A linear cut $\mathcal{V}_{t-1}$, an over-estimate v_{t-1}, a state x_t, and a gap value γ_t as in Definition 2

1: **for** $n \in \mathcal{N}(t)$ **do**
2: Solve Lagrangian dual of $\min\left\{f_n(z_n, y_n, x_n) + \underline{\mathcal{Q}}_t(x_n) + M_{t-1}\|x_{t-1} - z_n\| : (x_n, y_n) \in \mathcal{F}_n\right\}$
3: Collect the primal-dual solution pair $(\hat{x}_n, \hat{y}_n, \hat{z}_n; \hat{\lambda}_n)$ and the value $\underline{v}_n$
4: Define $V_n(x) \coloneqq \underline{v}_n + \langle \hat{\lambda}_n, x - x_{t-1}\rangle$
5: Calculate $\gamma_n \coloneqq \overline{\mathcal{Q}}_t(x_n) - \underline{\mathcal{Q}}_t(x_n)$
6: Solve for the primal value $\bar{v}_n$ of $\min\left\{f_n(z_n, y_n, x_n) + \overline{\mathcal{Q}}_t(x_n) + M_{t-1}\|x_{t-1} - z_n\| : (x_n, y_n) \in \mathcal{F}_n\right\}$
7: **end for**
8: Construct $\mathcal{V}_{t-1}(x) \coloneqq \sum_{n\in\mathcal{N}(t)} \hat{p}_{t-1,n} V_n(x)$ where $\hat{p}_{t-1} \in \arg\max_{p_{t-1}\in\mathcal{P}_{t-1}} \sum_{n\in\mathcal{N}} p_{t-1,n}\underline{v}_n$
9: Calculate $v_{t-1} \coloneqq \max_{p_{t-1}\in\mathcal{P}_{t-1}} p_{t-1,n}\bar{v}_n$
10: Find $n^* \in \mathcal{N}(t)$ such that $\gamma_{n^*} \ge \gamma_n$ for all $n \in \mathcal{N}(t)$ and set $x_t \coloneqq x_{n^*}$, $\gamma_t \coloneqq \gamma_{n^*}$

3.2 Dual Dynamic Programming Algorithms

With the subproblem oracles, we first introduce a sequential dual dynamic programming (Seq-DDP) algorithm. To ease the notation, we use $g = \mathrm{conv}\{h_1, h_2\}$ to denote the function corresponding to the closed convex hull of the epigraphs of functions h_1 and h_2. More precisely, using convex bi-conjugacy, we define

$$g(x) \coloneqq \left(\min\{h_1(x), h_2(x)\}\right)^{**} = \sup_{\lambda}\inf_{z}\left\{\min\{h_1(z), h_2(z)\} + \langle\lambda, x - z\rangle\right\}.$$

Note that if h_1, h_2 are both polyhedral (hence closed and convex), then by linear program strong duality, the function g can be represented as

$$g(x) = \min\left\{v \in \mathbb{R} : (z, v) = \mu_1(z_1, v_1) + \mu_2(z_2, v_2), v_i \ge h_i(z_i), \mu_i \ge 0, i = 1, 2, \mu_1 + \mu_2 = 1\right\},$$

assuming $g(x)$ is proper.

For each iteration $i \in \mathbb{N}$, the main loop of Algorithm 2 consists of three parts. The forward step uses the state x_{t-1}^i in the previous stage and the approximations $\underline{\mathcal{Q}}_t^{i-1}$ and $\overline{\mathcal{Q}}_t^{i-1}$ to produce a new state x_t^i. Then the backward step at stage t uses the cut $\mathcal{V}_{t-1}^i(x)$ and the value v_{t-1}^i to update the approximations $\underline{\mathcal{Q}}_{t-1}^i, \overline{\mathcal{Q}}_{t-1}^i$ in its precedent stage $t-1$. Finally, the initial stage step produces a new first stage solution (x_1^{i+1}, y_1^{i+1}) and updates the lower and upper bounds.

We next show the correctness of Algorithm 2, i.e., the returned solution (x_1^*, y_1^*) is ε-optimal, while leaving the finiteness proof to Section 3.3. From the termination of the while-loop, it suffices to show that the approximations are valid $\underline{\mathcal{Q}}_t^i(x) \le \mathcal{Q}_t^{\mathrm{R}}(x) \le \overline{\mathcal{Q}}_t^i(x)$

Algorithm 2 Sequential Dual Dynamic Programming Algorithm

Require: Subproblem oracles $\mathscr{O}_t$ for $t \in \mathcal{T}$, optimality gap $\varepsilon > 0$

Ensure: An ε-optimal first stage solution (x_1^*, y_1^*) to the regularization (10)

Initialize: $\underline{\mathcal{Q}}_t^0 \leftarrow 0, \overline{\mathcal{Q}}_t^0 \leftarrow +\infty, t \in \mathcal{T}\backslash\{T\}$; $\underline{\mathcal{Q}}_T^j, \overline{\mathcal{Q}}_T^j \leftarrow 0, j \in \mathbb{N}$; $i \leftarrow 1$
Evaluate $(x_1^1, y_1^1; \gamma_1^1) = \mathscr{O}_1(\underline{\mathcal{Q}}_1^0, \overline{\mathcal{Q}}_1^0)$
Set LowerBound $\leftarrow f_1(x_0, y_1^1, x_1^1)$, UpperBound $\leftarrow +\infty$
while UpperBound $-$ LowerBound $> \varepsilon$ **do**
  for $t = 2, \ldots, T$ **do**
    Evaluate $(\mathcal{V}_{t-1}^i, v_{t-1}^i, x_t^i; \gamma_t^i) = \mathscr{O}_t(x_{t-1}^i, \underline{\mathcal{Q}}_t^{i-1}, \overline{\mathcal{Q}}_t^{i-1})$ ▷ Forward step
  end for
  for $t = T, \ldots, 2$ **do**
    Update $\underline{\mathcal{Q}}_{t-1}^i(x) \leftarrow \max\{\underline{\mathcal{Q}}_{t-1}^{i-1}(x), \mathcal{V}_{t-1}^i(x)\}$ ▷ Backward step
    Update $\overline{\mathcal{Q}}_{t-1}^i(x) \leftarrow \operatorname{conv}\{\overline{\mathcal{Q}}_{t-1}^{i-1}(x), v_{t-1}^i + M_{t-1}\left\|x - x_{t-1}^i\right\|\}$
  end for
  Evaluate $(x_1^{i+1}, y_1^{i+1}; \gamma_1^i) = \mathscr{O}_1(\underline{\mathcal{Q}}_1^i, \overline{\mathcal{Q}}_1^i)$ ▷ Initial stage step
  Update LowerBound $\leftarrow f_1(x_0, y_1^{i+1}, x_1^{i+1}) + \underline{\mathcal{Q}}_1^i(x_1^{i+1})$
  Update UpperBound$' \leftarrow f_1(x_0, y_1^{i+1}, x_1^{i+1}) + \overline{\mathcal{Q}}_1^i(x_1^{i+1})$
  if UpperBound$'$ $<$ UpperBound **then**
    Set $(x_1^*, y_1^*) \leftarrow (x_1^{i+1}, y_1^{i+1})$, UpperBound $\leftarrow$ UpperBound$'$
  end if
  Update $i \leftarrow i + 1$
end while

for each $t \in \mathcal{T}$ and $i \in \mathbb{N}$. The first inequality follows from the validness of linear cuts $V_t^i(x)$ (cf. Proposition 3). The second inequality is due to the M_t-Lipschitz continuity of the regularized cost-to-go functions $\mathcal{Q}_t^{\mathrm{R}}$. In particular, by Definition 2, whenever the input $\overline{\mathcal{Q}}_t^i(x) \geq \mathcal{Q}_t^{\mathrm{R}}(x)$ for $x \in \mathcal{X}_t$, the value $v_{t-1}^i \geq \mathcal{Q}_{t-1}^{\mathrm{R}}(x_{t-1}^i)$. Then $v_{t-1}^i + M_t\left\|x - x_{t-1}^i\right\| \geq \mathcal{Q}_t^{\mathrm{R}}(x)$ for all $x \in \mathcal{X}_{t-1}$. Given that $\overline{\mathcal{Q}}_{t-1}^{i-1}(x) \geq \mathcal{Q}_{t-1}^{\mathrm{R}}(x)$ for $x \in \mathcal{X}_{t-1}$, which is obviously true for $i = 1$, we conclude that

$$\min\{\overline{\mathcal{Q}}_{t-1}^{i-1}(x), v_{t-1}^i + M_{t-1}\|x - x_{t-1}^i\|\} \geq \mathcal{Q}_{t-1}^{\mathrm{R}}(x), \quad \forall\, x \in \mathcal{X}_{t-1}. \tag{28}$$

By taking the closed convex hull of the epigraphs on both sides, we have shown that $\overline{\mathcal{Q}}_{t-1}^i(x) \geq \mathcal{Q}_{t-1}^{\mathrm{R}}(x)$ for all $x \in \mathcal{X}_{t-1}$. The above argument shows inductively that for all $i \in \mathbb{N}$, the approximations are valid, which then implies the correctness of the algorithm.

We comment that the linear cut $\mathcal{V}_{t-1}^i$ and the over-estimate value v_{t-1}^i are generated using only the information in the previous iteration $i - 1$. In fact, the subproblem oracles can be re-evaluated in the backward steps to produce tighter approximations. We simply keep the Seq-DDP algorithm in its current form because it is already sufficient for us to provide its complexity bound. At the same time, we propose an alternative nonsequential version of the dual dynamic programming (NDDP) algorithm that could possibly conduct

more efficient approximation updates.

Algorithm 3 Nonsequential Dual Dynamic Programming Algorithm

Require: Subproblem oracles $\mathscr{O}_t$ for $t \in \mathcal{T}$, opt. and approx. gaps $\varepsilon = \delta_1 > \cdots > \delta_T = 0$
Ensure: An ε-optimal first stage solution (x_1^*, y_1^*) to the regularization (10)

Initialize: $\underline{\mathcal{Q}}_t^0 \leftarrow 0, \overline{\mathcal{Q}}_t^0 \leftarrow +\infty, t \in \mathcal{T}\backslash\{T\}$; $\underline{\mathcal{Q}}_T^j, \overline{\mathcal{Q}}_T^j \leftarrow 0, j \in \mathbb{N}$; $i_t \leftarrow 0, t \in \mathcal{T}$
Set LowerBound $\leftarrow 0$, UpperBound $\leftarrow +\infty$, $t \leftarrow 1$
while true do
  Update $i_t \leftarrow i_t + 1$
  if $t = 1$ **then**
    Evaluate $(x_1^{i_1}, y_1^{i_1}; \gamma_1^{i_1}) = \mathscr{O}_1(\underline{\mathcal{Q}}_1^{i_1}, \overline{\mathcal{Q}}_1^{i_1})$ ▷ Initial stage step
    Update LowerBound $\leftarrow f_1(x_0, y_1^{i_1}, x_1^{i_1}) + \underline{\mathcal{Q}}_1^{i_1}(x_1^{i_1})$
    Update UpperBound$'$ $\leftarrow f_1(x_0, y_1^{i_1}, x_1^{i_1}) + \overline{\mathcal{Q}}_1^{i_1}(x_1^{i_1})$
    if UpperBound$'$ $<$ UpperBound **then**
      Set $(x_1^*, y_1^*) \leftarrow (x_1^{i_1}, y_1^{i_1})$, UpperBound $\leftarrow$ UpperBound$'$
    end if
    if UpperBound $-$ LowerBound $\leq \varepsilon$ **then**
      break
    end if
    Maintain $\underline{\mathcal{Q}}_2^{i_2+1}(x) \leftarrow \underline{\mathcal{Q}}_2^{i_2}(x)$, $\overline{\mathcal{Q}}_2^{i_2+1}(x) \leftarrow \overline{\mathcal{Q}}_2^{i_2}(x)$
    Set $t \leftarrow t + 1$
  else
    Evaluate $(\mathcal{V}_{t-1}^{i_t}, v_{t-1}^{i_t}, x_t^{i_t}; \gamma_t^{i_t}) = \mathscr{O}_t(x_{t-1}^{i_{t-1}}, \underline{\mathcal{Q}}_t^{i_t}, \overline{\mathcal{Q}}_t^{i_t})$ ▷ Noninitial stage step
    if $t < T$ **and** $\gamma_t^{i_t} > \delta_t$ **then**
      Maintain $\underline{\mathcal{Q}}_{t+1}^{i_{t+1}+1}(x) \leftarrow \underline{\mathcal{Q}}_{t+1}^{i_{t+1}}(x)$, $\overline{\mathcal{Q}}_{t+1}^{i_{t+1}+1}(x) \leftarrow \overline{\mathcal{Q}}_{t+1}^{i_{t+1}}(x)$
      Set $t \leftarrow t + 1$
    else
      Update $\underline{\mathcal{Q}}_{t-1}^{i_{t-1}+1}(x) \leftarrow \max\{\underline{\mathcal{Q}}_{t-1}^{i_{t-1}}(x), \mathcal{V}_{t-1}^{i_t}(x)\}$
      Update $\overline{\mathcal{Q}}_{t-1}^{i_{t-1}+1}(x) \leftarrow \mathrm{conv}\{\overline{\mathcal{Q}}_{t-1}^{i_{t-1}}(x), v_{t-1}^{i_t} + M_{t-1}\|x - x_{t-1}^{i_{t-1}}\|\}$
      Set $t \leftarrow t - 1$
    end if
  end if
end while

Algorithm 3 describes the NDDP algorithm. To start the algorithm, it requires an additionally chosen vector of approximation gaps $\delta := (\delta_t)_{t=1}^T$ such that $\varepsilon = \delta_1 > \delta_2 > \cdots > \delta_T = 0$, compared with the Seq-DDP algorithm. These predetermined approximation gaps serve as criteria at stage t for deciding the next stage to be solved: the precedent stage $t-1$ or the subsequent one $t+1$. If the algorithm decides to proceed to the subsequent stage $t+1$, then the current state x_t is used; otherwise the generated linear cut $\mathcal{V}_{t-1}$ and over-estimate value v_{t-1} are used for updating the approximations. The above argument of validness of approximations imply that $\underline{\mathcal{Q}}_t^{i_t}(x) \leq \mathcal{Q}_t^{\mathrm{R}}(x) \leq \overline{\mathcal{Q}}_t^{i_t}(x)$ for all $x \in \mathcal{X}_t$ holds for

any stage $t \in \mathcal{T}$ and any index $i_t \in \mathbb{N}$. Therefore, when NDDP terminates, the returned solution (x_1^*, y_1^*) is indeed ε-optimal.

3.3 Complexity Upper Bounds

In this section, we provide a complexity analysis for the proposed Seq-DDP and NDDP algorithms, which implies that both algorithms terminate in finite time. Our goal is to derive an upper bound on the total number of subproblem oracle evaluations before the termination of the algorithm. To begin with, let $\mathcal{J}_t, t > 1$ denote the set of pair of indices (i_{t-1}, i_t) such that the noninitial stage subproblem oracle is evaluated at the i_t-th time at the state $x_{t-1}^{i_{t-1}}$, i.e., $(\mathcal{V}_{t-1}^{i_t}, v_{t-1}^{i_t}, x_t^{i_t}; \gamma_t^{i_t}) = \mathscr{O}_t(x_{t-1}^{i_{t-1}}, \underline{\mathcal{Q}}_t^{i_t}, \overline{\mathcal{Q}}_t^{i_t})$. For the Seq-DDP algorithm, all stages share the same iteration index $i_t = i$, so $\mathcal{J}_t = \{(i,i) : i \in \mathbb{N}\}$ for all $t > 1$. We define the following sets of indices for each $t \in \mathcal{T} \setminus \{T\}$:

$$\mathcal{I}_t(\delta) \coloneqq \left\{ i_t \in \mathbb{N} : \gamma_t^{i_t} > \delta_t \text{ and } \gamma_{t+1}^{i_{t+1}} \le \delta_{t+1},\ (i_t, i_{t+1}) \in \mathcal{J}_{t+1} \right\}. \tag{29}$$

Here, for NDDP algorithm, δ is the given approximation gap vector, while for Seq-DDP algorithm, $\delta = (\delta_t)_{t=1}^T$ can be any vector satisfying $\varepsilon = \delta_1 > \delta_2 > \cdots > \delta_T = 0$ for the purpose of analysis, since it is not required for the Seq-DDP algorithm. We adopt the convention that the gap for the last stage $\gamma_T^{i_T} \equiv 0$ such that $i_{T-1} \in \mathcal{I}_{T-1}(\delta)$ if and only if $\gamma_{T-1}^{i_{T-1}} > \delta_{T-1}$ and $(i_{T-1}, i_T) \in \mathcal{J}_T$. An important observation is that all these index sets are finite (before algorithm termination) $|\mathcal{I}_t| < \infty$, which is more precisely stated in the following lemma.

Lemma 2. *For stage t, suppose the state space $\mathcal{X}_t \subset \mathbb{R}^{d_t}$ is contained in a ball with diameter $D_t > 0$. Then,*

$$\left|\mathcal{I}_t(\delta)\right| \le \left(1 + \frac{2M_t D_t}{\delta_t - \delta_{t+1}}\right)^{d_t}. \tag{30}$$

Proof. We claim that for any $j, k \in \mathcal{I}_t$, $j \ne k$, it holds that $\|x_t^j - x_t^k\| > (\delta_t - \delta_{t+1})/(2M_t)$. Assume for contradiction that $\|x_t^j - x_t^k\| \le (\delta_t - \delta_{t+1})/(2M_t)$ for some $j < k$, $j, k \in \mathcal{I}_t(\delta)$. By definition of $\mathcal{I}_t(\delta)$, the $t+1$-th subproblem oracle is evaluated at the state x_t^j, and in both the Seq-DDP and the NDDP algorithms, the approximations $\underline{\mathcal{Q}}_t^j$ and $\overline{\mathcal{Q}}_t^j$ are updated since $\gamma_{t+1}^{i_{t+1}} \le \delta_{t+1}$ for some $i_{t+1} \in \mathbb{N}$ with $(j, i_{t+1}) \in \mathcal{J}_{t+1}$. Then by Definition 2 of the noninitial stage subproblem oracle, we have $\overline{\mathcal{Q}}_t^j(x_t^j) - \underline{\mathcal{Q}}_t^j(x_t^j) \le \delta_{t+1}$. Following Proposition 3, for any point $x \in \mathcal{X}_t$ with $\|x - x_t^j\| \le (\delta_t - \delta_{t+1})/(2M_t)$, we have $\overline{\mathcal{Q}}_t^j(x) - \underline{\mathcal{Q}}_t^j(x) \le \delta_t$ because of the M_t-Lipschitz continuity of the approximations. By setting $x = x_t^k$, we see a contradiction with the assumption that $k \in \mathcal{I}_t(\delta)$, which proves the claim.

To ease the notation, let $r_t \coloneqq (\delta_t - \delta_{t+1})/(2M_t)$ denote the radius of the d_t-dimensional balls $\mathcal{B}^{d_t}(x_t^j; r_t)$ centered at x_t^j for $j \in \mathcal{I}_t(\delta)$, and let $\mathcal{B}_t \supseteq \mathcal{X}_t$ denote a ball with diameter D_t. From the above claim, we know that $x_t^k \notin \mathcal{B}^{d_t}(x_t^j; r_t)$ for any $j, k \in \mathcal{I}_t(\delta)$ with $j < k$. In other words, the smaller balls $\mathcal{B}^{d_t}(x_t^j; r_t/2)$ are disjoint. Meanwhile, note that each of these

smaller balls satisfies $\mathcal{B}^{d_t}(x_t^j; r_t/2) \subset \mathcal{B}_t + \mathcal{B}^{d_t}(0; r_t/2)$ (the Minkowski sum in the Euclidean space $\mathbb{R}^{d_t}$). Therefore, the volumes satisfy the relation

$$\mathrm{Vol}\left(\bigcup_{j\in\mathcal{I}_t(\delta)} \mathcal{B}^{d_t}(x_t^j; r_t/2)\right) = \left|\mathcal{I}_t(\delta)\right| \cdot \mathrm{Vol}\mathcal{B}^{d_t}(0; r_t/2) \leq \mathrm{Vol}\left(\mathcal{B}_t + \mathcal{B}^{d_t}(0; r_t/2)\right),$$

which implies that

$$\left|\mathcal{I}_t(\delta)\right| \leq \frac{\mathrm{Vol}(\mathcal{B}_t + \mathcal{B}^{d_t}(0; r_t/2))}{\mathrm{Vol}\mathcal{B}^{d_t}(0; r_t/2)} = \left(\frac{D_t/2 + r_t/2}{r_t/2}\right)^{d_t} = \left(1 + \frac{2M_tD_t}{\delta_t - \delta_{t+1}}\right)^{d_t}.$$

Thus we complete the proof. □

We prove the following complexity upper bounds for the Seq-DDP algorithm (Theorem 1) and the NDDP algorithm (Theorem 2).

Theorem 1. *Suppose the state spaces $\mathcal{X}_t \subset \mathbb{R}^{d_t}$ are contained in balls, each with diameter $D_t > 0$. Then for the Seq-DDP algorithm (Algorithm 2), the total number of subproblem oracle evaluations $\texttt{\#Eval}_{\text{Seq-DDP}}$ before termination is bounded by*

$$\texttt{\#Eval}_{\text{Seq-DDP}} \leq 1 + T \cdot \inf_{\delta} \left\{ \sum_{t=1}^{T-1} \left(1 + \frac{2M_tD_t}{\delta_t - \delta_{t+1}}\right)^{d_t} : \varepsilon = \delta_1 > \delta_2 > \cdots > \delta_T = 0 \right\}.$$

Proof. We prove by showing that for any approximation gap vector δ satisfying $\varepsilon = \delta_1 > \delta_2 > \cdots > \delta_T = 0$, the largest iteration index i is bounded by

$$i \leq \sum_{t=1}^{T-1} \left(1 + \frac{2M_tD_t}{\delta_t - \delta_{t+1}}\right)^{d_t}. \tag{31}$$

We claim that each iteration $i \in \mathbb{N}$ must lie in either of the following two cases:

1. the initial stage step has $\gamma_1^i \leq \varepsilon$; or
2. the i-th forward step is in the index set $i \in \mathcal{I}_t(\delta)$ for some stage $t < T$.

To see the claim, suppose that the iteration $i \in \mathbb{N}$ is not in the first case. Then we have $\gamma_1^i > \varepsilon$ and by convention $\gamma_T^i = 0 \leq \delta_T$. Therefore, there exists a stage $t < T$ such that $\gamma_t^i > \delta_t$ while $\gamma_{t+1}^i \leq \delta_{t+1}$, which is the second case. Note that when the first case happens, we have $\textsc{UpperBound} - \textsc{LowerBound} \leq \gamma_1^i \leq \varepsilon$ and thus the Seq-DDP algorithm terminates. By Lemma 2, the second case can only happen at most $\sum_{t=1}^{T-1}\left|\mathcal{I}_t(\delta)\right|$ times, proving the bound (31). The theorem then follows from the fact that in each Seq-DDP iteration, the subproblem oracle is evaluated T times and one additional evaluation of the initial stage subproblem oracle is needed for checking the termination criterion. □

Theorem 2. *Suppose the state spaces $\mathcal{X}_t \subset \mathbb{R}^{d_t}$ are contained in balls, each with diameter $D_t > 0$. Then, for the NDDP algorithm (Algorithm 3) with the predetermined approximation gap vector $(\delta_t)_{t=1}^T$ satisfying $\varepsilon = \delta_1 > \delta_2 > \cdots > \delta_T = 0$, the total number of subproblem oracle evaluations $\texttt{\#Eval}_{\text{NDDP}}$ before termination is bounded by*

$$\texttt{\#Eval}_{\text{NDDP}} \leq 1 + 2 \cdot \sum_{t=1}^{T-1} \left(1 + \frac{2M_tD_t}{\delta_t - \delta_{t+1}}\right)^{d_t}.$$

Proof. For the NDDP algorithm, each time when it decides to go back to the precedent stage $t \leftarrow t-1$, we must have $\gamma_t^{i_t} \leq \delta_t$ while $\gamma_{t-1}^{i_{t-1}} > \delta_{t-1}$ for some $(i_{t-1}, i_t) \in \mathcal{J}_t$. In this case, we have by definition that $i_t \in \mathcal{I}_t(\delta)$. By Lemma 2, such "going back" step can only happen at most $\sum_{t=1}^{T-1} |\mathcal{I}_t(\delta)|$ times. The theorem then follows from the fact that there are exactly two times such "going back" cases and one additional evaluation of the single stage subproblem oracle for checking the termination criterion. □

Let us compare the complexity bounds of the two algorithms. If we fix the approximation gap vector δ in Theorem 1 to be the same in the NDDP algorithm, then the complexity bound of Seq-DDP is worse than that of NDDP as $T \geq 2$. However, since an optimal choice of the gap vector δ is usually not known, Seq-DDP has the advantage of not requiring an a-priori estimate of these factors for the complexity bound to be valid. We provide below an important simplification of the above complexity bounds that applies to many practical problems.

Corollary 1. *Suppose that all the state spaces have the same dimension $d_t = d$ and bounded by a common diameter $D_t \leq D$, and let $M := \max\{M_t : t = 1, \ldots, T-1\}$. If for each stage $t \in \mathcal{T}$, the local cost functions are strictly positive for all feasible solutions $f_n(x_{t-1}, y_n, x_n) \geq C$, $n \in \mathcal{N}(t)$ for some $C > 0$, then the total number of subproblem oracle evaluations before achieving an α-relative optimal solution (x_1^*, y_1^*) for Seq-DDP and NDDP are upper bounded respectively by*

$$\texttt{\#Eval}_{\text{Seq-DDP}} \leq 1 + T(T-1)\left(1 + \frac{2MD}{\alpha C}\right)^d, \quad \texttt{\#Eval}_{\text{NDDP}} \leq 1 + 2(T-1)\left(1 + \frac{2MD}{\alpha C}\right)^d.$$

Proof. Note that if an solution (x_1^*, y_1^*) is ε-optimal with $\varepsilon = \alpha C(T-1) < \alpha CT$, then it is also α-relative optimal. The result then follows from Theorems 1 and 2 by setting $\delta_t = (T-t)\alpha C$. □

Corollary 1 shows that for problems that have strictly positive cost in each stage, the proposed complexity bounds for an α-relative optimal solution grow at most *quadratically* for Seq-DDP and *linearly* for NDDP with respect to the number of stages T. This provides an answer to the open question about the iteration complexity of DDP-type algorithms. In the next subsection, we will show this complexity bound is essentially tight by providing a matching lower bound.

3.4 Complexity Lower Bound

Note that if we take $\delta_t = \varepsilon(T-t)/(T-1)$ for $t \in \mathcal{T}$, then the complexity upper bounds in Theorems 1 and 2 depend on the terms $(T-1)^{d_t}$ where d_t is the state space dimension of stage $t < T$. It is natural to ask whether it is possible for either algorithm to achieve an ε-optimal solution with complexity that is linear or quadratic in T, independent of the state space dimensions (cp. Corollary 1). We present a class of convex problems to show that this is indeed *impossible*.

Given a d-sphere $\mathcal{S}^d(r) = \{x \in \mathbb{R}^{d+1} : \|x\|_2 = r\}$ with radius $r > 0$, a spherical cap with depth $\theta > 0$ centered at a point $x \in \mathcal{S}^d(r)$ is the set $\mathcal{S}^d_\theta(r, x) \coloneqq \{y \in \mathcal{S}^d(r) : \langle y - x, x\rangle \geq -\theta r\}$. The next lemma shows that we can put many spherical caps on a sphere such that the center of each is not contained in any other spherical cap. This is a key technical result needed for proving lower complexity bound. Let $\Gamma(\cdot)$ denote the gamma function.

Lemma 3. *Given a d-sphere $\mathcal{S}^d(r), d \geq 2$ and depth $\theta < (1 - \frac{\sqrt{2}}{2})r$, there exists a finite set of points $\mathcal{W}$ with*

$$|\mathcal{W}| \geq \frac{(d^2-1)\sqrt{\pi}}{d} \frac{\Gamma(d/2+1)}{\Gamma(d/2+3/2)} \left(\frac{r}{2\theta}\right)^{(d-1)/2},$$

such that for any $w \in \mathcal{W}$, $\mathcal{S}^d_\theta(r, w) \cap \mathcal{W} = \{w\}$.

Proof. Let v_d denote the d-volume for a d-dimensional unit ball. Recall that the d-volume for $\mathcal{S}^d(r)$ is given by $\mathrm{Vol}_d(\mathcal{S}^d(r)) = (d+1)v_{d+1}r^d = \frac{(d+1)\pi^{(d+1)/2}}{\Gamma(\frac{d+1}{2}+1)} r^d$. We next estimate the d-volume for the spherical cap $\mathcal{S}^d_\theta(r, x)$. Let $\alpha \in (0, \pi/2)$ denote the central angle for the spherical cap, i.e., $\cos\alpha = 1 - \theta/r$. Since $\theta < (1 - \frac{\sqrt{2}}{2})r$, we know that $\alpha < \pi/4$. Then for any $x \in \mathcal{S}^d(r)$, the d-volume of the spherical cap can be calculated through

$$\mathrm{Vol}_d(\mathcal{S}^d_\theta(r, x)) = \int_0^\alpha \mathrm{Vol}_{d-1}(\mathcal{S}^{d-1}(r\sin\varphi)) r \,\mathrm{d}\varphi = dv_d r^d \int_0^\alpha (\sin\varphi)^{d-1} \,\mathrm{d}\varphi.$$

Note that when $\varphi \in (0, \alpha)$, $\sin\varphi > 0$ and $\cos\varphi/\sin\varphi > 1$. Therefore, since $d \geq 2$,

$$\mathrm{Vol}_d(\mathcal{S}^d_\theta(r, x)) \leq dv_d r^d \int_0^\alpha (\sin\varphi)^{d-1} \frac{\cos\varphi}{\sin\varphi} \,\mathrm{d}\varphi = dv_d r^d \cdot \frac{(\sin\alpha)^{d-1}}{d-1}.$$

By substituting $\sin\alpha = \sqrt{1 - (1-\theta/r)^2}$, we have

$$\frac{\mathrm{Vol}_d(\mathcal{S}^d_\theta(r, x))}{\mathrm{Vol}_d(\mathcal{S}^d(r))} \leq \frac{d}{d^2-1}\frac{v_d}{v_{d+1}}(\sin\alpha)^{d-1} = \frac{d}{d^2-1}\frac{v_d}{v_{d+1}}\big(1 - \big(1 - \frac{\theta}{r}\big)^2\big)^{(d-1)/2} \leq \frac{d}{d^2-1}\frac{v_d}{v_{d+1}}\left(\frac{2\theta}{r}\right)^{(d-1)/2}.$$

Now suppose $\mathcal{W} = \{w_i\}_{k=1}^K$ is a maximal set satisfying the assumption, that is, for any $w \in \mathcal{S}^d(r), w \notin \mathcal{W}$, there exists $w_k \in \mathcal{W}$ such that $w \in \mathcal{S}^d_\theta(r, w_k)$. Then, $\cup_{k=1}^K \mathcal{S}^d_\theta(r, w_k) \supseteq \mathcal{S}^d(r)$, and thus $\mathrm{Vol}_d(\mathcal{S}^d(r)) \leq \sum_{k=1}^K \mathrm{Vol}_d(\mathcal{S}^d_\theta(r, w_k)) = |\mathcal{W}| \,\mathrm{Vol}_d(\mathcal{S}^d_\theta(r, w_1))$. Therefore we have

$$|\mathcal{W}| \geq \frac{\mathrm{Vol}_d(\mathcal{S}^d(r))}{\mathrm{Vol}_d(\mathcal{S}^d_\theta(r, w_1))} \geq \left[\frac{d}{d^2-1}\frac{v_d}{v_{d+1}}\left(\frac{2\theta}{r}\right)^{(d-1)/2}\right]^{-1} = \frac{(d^2-1)\sqrt{\pi}}{d}\frac{\Gamma(d/2+1)}{\Gamma(d/2+3/2)}\left(\frac{r}{2\theta}\right)^{(d-1)/2}.$$

This completes the proof. □

We denote the set of points associated to the sphere $\mathcal{S}^d(r)$ as $\mathcal{W}_\theta^d(r) = \{w_k\}_{k=1}^K$ in Lemma 3. For any constants $\varepsilon > 0$ and $l > 0$, and values $v_k \in (\varepsilon/2, \varepsilon)$, $k = 1, \dots, K$, we define a function associated with the pair $(\mathcal{W} = \mathcal{W}_\theta^d(r), v = (v_k)_{k=1}^K)$ as $F^{\mathcal{W},v}(x) := \max_k\{0, v_k + \frac{l}{r}\langle w_k, x - w_k\rangle\}$, $x \in \mathcal{B}^{d+1}(r)$, which is obviously convex and l-Lipschitz continuous. Moreover, if the depth θ and the constants ε, l satisfy $l\theta \geq \varepsilon$, then we have $v_k + \frac{l}{r}\langle w_k, w_{k'} - w_k\rangle < v_k - \frac{l}{r}\theta r \leq v_k - \varepsilon < 0$ for any $w_{k'} \neq w_k \in \mathcal{W}_\theta^d(r)$. This implies that $F^{\mathcal{W},v}(w_k) = v_k$ and the subdifferential $\partial F^{\mathcal{W},v}(w_k) = \{\frac{l}{r} w_k\}$ for all $k = 1, \dots, K$. Another important observation is that the convex function $\overline{F}^{\neq k}(x) := \mathrm{conv}_{k' \neq k}\{v_{k'} + l\,\|x - w_{k'}\|\}$ has the property that $\overline{F}^{\neq k}(w_k) > \varepsilon/2 + l \cdot \mathrm{dist}(w_k, \mathrm{conv}_{k' \neq k}\{w_{k'}\}) \geq \varepsilon/2 + l\theta \geq 3\varepsilon/2$.

We next construct a class of MRCO's using such convex functions, with the following parameters: $T \geq 3$ as the number of stages, $L > 0$ as a prescribed Lipschitz constant, $d \geq 3$ as the state space dimension, $D = 2r > 0$ as the state space diameter, and $\varepsilon > 0$ as the optimality gap. Choose any $l_1, \dots, l_{T-1}$ such that $L/2 = l_T < l_{T-1} < \cdots < l_1 = L$, and set $\varepsilon_t = 2\varepsilon/(T-2)$. Construct sets of points $\mathcal{W}_t := \mathcal{W}_{\theta_t}^{d-1}(r) = \{w_{t,k}\}_{k=1}^{K_t}$, where $\theta_t = \varepsilon_t / l_t$ for $t = 1, \dots, T-1$. Let $F_t(x) = F^{\mathcal{W}_t, v_t}(x)$ be constructed as above for any values $v_t = (v_{t,k})_{k=1}^{K_t}$, $v_{t,k} \in (\varepsilon_t/2, \varepsilon_t)$, and the Lipschitz constant l_t, $k = 1, \dots, K_t$, for $t > 1$ and $F_1(x) \equiv 0$. The problem is then constructed as

$$\mathcal{Q}_t(x_t) = \max_{\xi_t \in \mathrm{conv}(\mathcal{W}_{t+1})} \min_{x_{t+1} \in \mathcal{B}^d(r)} \left\{F_t(x_t) + l_t\,\|x_{t+1} - \xi_t\| + \mathcal{Q}_{t+1}(x_{t+1})\right\}, 1 < t < T, \quad (32)$$

where $\mathcal{Q}_T(x) \equiv 0$, and the deterministic first stage problem is defined as $\min_{x_1 = 0} \mathcal{Q}_1(x_1) = \mathcal{Q}_1(0)$. In other words, this class of problems seeks the optimal value corresponding to $x_1 = 0$. We are now at the point to give the lower bound of the complexity of Seq-DDP and NDDP algorithms, assuming the regularization factors $M_t \geq L$, $t \in \mathcal{T}$ for the exactness by Proposition 4.

Theorem 3. *For the problem (32), the number of subproblem oracle evaluations* `#Eval` *for either of Algorithms 2 and 3 before termination has the following lower bound*

$$\texttt{\#Eval} \geq \frac{d(d-2)\sqrt{\pi}}{d-1} \frac{\Gamma(d/2 + 1/2)}{\Gamma(d/2+1)} \left(\frac{DL(T-2)}{16\varepsilon}\right)^{(d-2)/2} = \mathcal{O}(T^{d/2-1}) \text{ as } T \to \infty.$$

Proof. By reformulation of the problem (32), we assume that the algorithms only consider the worst-case uncertainty vector $\xi_t \in \mathcal{W}_t$. Note that for $1 < t < T$,

$$\mathcal{Q}_t(x_t) = F_t(x_t) + \max_{\xi_t \in \mathcal{W}_{t+1}} \min_{x_{t+1} \in \mathcal{B}^d(r)} \{l_t\,\|x_{t+1} - \xi_t\| + \mathcal{Q}_{t+1}(x_{t+1})\} =: F_t(x_t) + c_{t+1}.$$

Therefore, the cost-to-go function $\mathcal{Q}_t$ and any under-approximation $\underline{\mathcal{Q}}_t$ is l_t-Lipschitz continuous, which means $\mathcal{Q}_t^{\mathrm{R}} = \mathcal{Q}_t$ by Proposition 4. We further assume that the return of the SSSO satisfies $x_t = \xi_{t-1} \in \mathcal{W}_{t-1}$, which is true for the case of Algorithm 1 since $l_{t-1} > l_t$ implies $x_t = \xi_{t-1}$ is the unique minimizer to the recursion (32).

Now let $\underline{\mathcal{Q}}_t^{i_t}, \overline{\mathcal{Q}}_t^{i_t}$ denote the under- and over-approximations of $\mathcal{Q}_t$ at stage index i_t, and $\bar{c}_t^{i_t}, \underline{c}_t^{i_t}$ denote the corresponding under- and over-estimations of the value c_t. Let

$\underline{F}_t^{i_t}(x) \coloneqq \max\{0, \max\{v_{t,k} + \frac{l_t}{r}\langle w_{t,k}, x - w_{t,k}\rangle : w_{t,k} = x_t^{i_t}, (i_t, i_{t+1}) \in \mathcal{J}_{t+1}\}\}$ and $\overline{F}_t^{i_t}(x) \coloneqq \operatorname{conv}\{v_{t,k} + M_t \|x - w_{t,k}\| : w_{t,k} = x_t^{i_t}, (i_t, i_{t+1}) \in \mathcal{J}_{t+1}\}$ denote the under- and over-approximations of the function F_t. Note that $\underline{\mathcal{Q}}_t^{i_t}(x) \le \underline{F}_t^{i_t}(x) + \underline{c}_{t+1}^{i_{t+1}}$ and $\overline{\mathcal{Q}}_t^{i_t}(x) \ge \overline{F}_t^{i_t}(x) + \bar{c}_{t+1}^{i_{t+1}}$, for all $t \in \mathcal{T}$. For all $i_t < K_t$, there exists $w_{t,k} \in \mathcal{W}_t$ such that $\overline{F}_t^{i_t}(w_{t,k}) > 3\varepsilon_t/2$ from the discussion following the definition. Therefore, if $i_t < K_t$ for all $t > 1$, then we have

$$\begin{aligned}\bar{c}_t^{i_t} - \underline{c}_t^{i_t} &= \max_{\xi_{t-1}\in\mathcal{W}_t} \min_{x_t\in\mathcal{B}^d(r)} \{l_{t-1}\|x_t - \xi_{t-1}\| + \overline{\mathcal{Q}}_t^{i_t}(x_t)\} - \max_{\xi_{t-1}\in\mathcal{W}_t} \min_{x_t\in\mathcal{B}^d(r)} \{l_{t-1}\|x_t - \xi_{t-1}\| + \underline{\mathcal{Q}}_t^{i_t}(x_t)\} \\ &\ge \overline{F}_t^{i_t}(w_{t,k}) - \max_{\xi_{t-1}\in\mathcal{W}_t} F_t(\xi_{t-1}) + \bar{c}_{t+1}^{i_{t+1}} - \underline{c}_{t+1}^{i_{t+1}} > \frac{\varepsilon_t}{2} + \bar{c}_{t+1}^{i_{t+1}} - \underline{c}_{t+1}^{i_{t+1}}.\end{aligned}$$

Therefore, UpperBound $-$ LowerBound $= \bar{c}_1^{i_1} - \underline{c}_1^{i_1} > \sum_t \varepsilon_t/2 \ge (T-2)\cdot 2\varepsilon/(T-2) = \varepsilon$. Equivalently, when the algorithms terminate, we must have $i_t \ge K_t$ for some $t > 1$, which implies

$$\begin{aligned}\texttt{\#Eval} \ge K_t &\ge \frac{((d-1)^2-1)\sqrt{\pi}}{d-1}\frac{\Gamma((d-1)/2+1)}{\Gamma((d-1)/2+3/2)}\left(\frac{rL_t(T-2)}{2\varepsilon_t}\right)^{(d-2)/2} \\ &\ge \frac{d(d-2)\sqrt{\pi}}{d-1}\frac{\Gamma(d/2+1/2)}{\Gamma(d/2+1)}\left(\frac{DL(T-2)}{16\varepsilon}\right)^{(d-2)/2},\end{aligned}$$

by Lemma 3 since $L_t \ge L/2$. This completes the proof. □

4 Numerical Tests

In this section, we numerically test the proposed Seq-DDP and NDDP algorithms. The first test problem is a robust multi-commodity inventory problem with customer demand uncertainty. The second test problem is a distributionally robust hydro-thermal power planning problem with stochastic energy inflows. The computation budget consists of 40 2.1-GHz CPU cores and a total of 80 GBytes of RAM. The algorithms are implemented using JuMP package ([11], v0.21) in Julia language (v1.4) with Gurobi 9.0 as its underlying LP solver.

4.1 Multi-Commodity Inventory Problem

We consider a multi-commodity inventory problem with uncertain customer demands and deterministic holding and backlogging costs, following the description in [15]. Due to the stagewise independence of the bounded uncertainties and convexity of the problem, we formulate the problem below as an MRCO recursion (6). Let $\mathcal{K} \coloneqq \{1, 2, \ldots, K\}$ denote the set of product indices. We first describe the variables in each stage $t \in \mathcal{T}$. We use $l_{t,k}$ to denote the inventory level, $a_{t,k}$ (resp. $b_{t,k}$) to denote the amount of express (resp. standard) order fulfilled in the current (resp. subsequent) stage, of some product $k \in \mathcal{K}$. Let $\xi_t \in \Xi_t$ denote the uncertainty vector controlling the customer demands in stage t. The first stage

is assumed to be deterministic, i.e., $\Xi_1 = \{0\}$ without loss of generality. Then, the stage t subproblem can be written as

$$\begin{aligned}
\mathcal{Q}_{t-1}(x_{t-1}) \coloneqq \max_{\xi_t \in \Xi_t} \min \quad & \sum_{k\in\mathcal{K}} \left(c^F + c^a_k a_{t,k} + c^b_k b_{t,k} + c^H_k [l_{t,k}]_+ + c^B_k [l_{t,k}]_- \right) + \mathcal{Q}_t(x_t) && (33)\\
\text{s.t.} \quad & \sum_{k\in\mathcal{K}} a_{t,k} \le B^c, \\
& l_{t,k} - a_{t,k} - b_{t-1,k} = l_{t,k} - d_{t,k}(\xi_t), && \forall\, k \in \mathcal{K}, \\
& a_{t,k} \in [0, B^a_k], && \forall\, k \in \mathcal{K}, \\
& b_{t,k} \in [0, B^b_k], && \forall\, k \in \mathcal{K}, \\
& l_{t,k} \in [-B^l_k, B^l_k], && \forall\, k \in \mathcal{K}.
\end{aligned}$$

Here in the formulation, c^a_k (resp. c^b) denotes the express (resp. standard) order unit cost, c^H_k (resp. c^B_k) the inventory holding (resp. backlogging) unit cost, B^a_k (resp. B^b_k) the productwise bound for the express (resp. standard) order, and B^l_k the inventory level bound, for the product k, respectively. The first constraint in (33) is a cumulative bound B^c on the express orders, the second constraint characterizes the change in the inventory level, and the rest are bounds on the decision variables with respect to each product. We also put $c^F > 0$ as a fixed cost to ensure the cost function is strictly positive (cf. Corollary 1). We use $[l]_+ \coloneqq \max\{l, 0\}$ and $[l]_- \coloneqq -\min\{0, l\}$ to denote the positive and negative part of a real number l. The state variables x_t consist of the inventory levels $(l_{t,k})_{k\in\mathcal{K}}$ and the standard order amounts $(b_{t,k})_{k\in\mathcal{K}}$, while the internal variables are the express order amounts $y_t = (a_{t,k})_{k\in\mathcal{K}}$. The initial state x_0 is given by $l_{0,k} = b_{0,k} = 0$ for all $k \in \mathcal{K}$. The uncertainty set Ξ_t is a E-dimensional box $[-1, 1]^E$, and the customer demand is predicted by the following factor model:

$$d_{t,k}(\xi_t) = \begin{cases} 2 + \sin\left(\dfrac{(t-1)\pi}{5}\right) + \Phi_{t,k}\xi_t, & k \le K/2, \\ 2 + \cos\left(\dfrac{(t-1)\pi}{5}\right) + \Phi_{t,k}\xi_t, & k > K/2, \end{cases} \qquad (34)$$

where $\Phi_{t,}$ is a E-dimensional vector where each entry is chosen uniformly at random from $[-1/E, 1/E]$. Thus the value $\Phi_{t,k}\xi_t \in [-1, 1]$ and $d_{t,k}(\xi_t) \ge 0$ for all $t \in \mathcal{T}$ and $k \in \mathcal{K}$.

For the following numerical test, we set the number of products $K = 5$, the number of uncertainty factors $E = 4$, $B^a_k = B^b_k = B^l_k = 10$, $c^b_k = 1$ for all $k \in \mathcal{K}$, $c^F = 1$, and $B^c = 0.3K = 1.5$. The costs are generated uniformly at random within $c^a_k \in [1, 3]$, $c^H_k, c^B_k \in [0, 2]$ for all $k \in \mathcal{K}$. Due to lack of relatively complete recourse of the problem (33), we use the nonsequential dual dynamic programming algorithm (Algorithm 3) with the the optimality gap set to be relative $\alpha = 1\%$ and approximation gaps set dynamically by $\delta^{i_t}_t = \textsc{LowerBound} \cdot \alpha(T-t)/(T-1)$ for $t \in \mathcal{T}$. As a comparison, we implement the same algorithm without regularization which generates linear feasibility cuts for approximation of the feasible sets (see definition of feasibility cuts in, e.g., [17]). For 5 independently

generated test cases, we have obtained the following results (Table 1) within a time limit of 5 hours and regularization factor of $M_t = 1.0 \times 10^2$ for all $t \in \mathcal{T}$.

	Regularized Problem				Unregularized Problem			
Stage	LB	UB	Time (s)	#Eval	LB	UB	Time (s)	#Eval
	154.66	155.92	203.99	1497	154.70	154.70	96.15	1448
	155.94	157.01	175.12	1502	155.95	inf	18000.29	14151
10	128.31	129.39	230.47	1638	127.56	inf	18002.84	14195
	137.01	138.37	168.36	1405	137.06	137.16	82.40	1565
	120.13	121.01	365.65	1948	120.13	121.16	160.65	2302
	232.46	234.35	473.59	3158	232.46	232.59	265.05	3109
	233.37	235.40	509.25	3107	233.44	235.18	121.63	2752
15	202.48	203.84	413.36	3038	201.89	inf	18001.74	15956
	208.59	209.78	443.00	2930	208.57	208.95	214.33	2992
	195.16	196.97	680.97	4272	187.11	inf	18069.06	26686
	291.87	294.41	1222.93	6189	291.90	292.76	404.01	5175
	292.47	294.87	1248.34	6156	292.84	293.77	437.24	4848
20	256.28	258.53	1026.97	5190	256.30	257.20	401.07	5519
	261.70	264.31	775.68	4770	261.81	261.81	385.10	4765
	249.86	251.32	2135.50	8455	248.12	inf	18072.89	26618
	369.52	370.89	3079.75	11507	369.66	370.31	613.28	7399
	370.01	373.55	2473.70	9814	370.47	370.87	529.98	7422
25	330.01	332.46	1474.33	7674	330.42	332.22	718.88	8291
	333.28	336.27	1009.98	6345	333.34	333.40	578.25	7247
	324.54	326.43	3283.15	12148	311.97	inf	18106.49	33727
	428.94	432.04	4319.78	14394	429.11	429.23	1004.61	10102
	429.48	430.88	3836.59	13707	407.22	inf	18057.45	29520
30	384.21	387.11	2445.64	11142	383.76	inf	18001.47	23233
	386.44	390.29	2006.87	10319	386.54	387.36	922.64	10277
	379.33	382.14	5981.94	16756	379.59	382.24	1615.91	15104

Table 1: Comparison of NDDP with and without regularization

In Table 1, the inf indicates values of infinity or numerically infinity values (i.e., values greater than 10^9) within the computation time of 5 hours. As we see from the table, the NDDP algorithm together with feasibility cuts fails to solve two out of five cases even when there is only 10 stages, showing the instability of the performance of feasibility cuts. In contrast, the algorithm with the regularization technique solves all of the cases within a reasonable computation time and number of subproblem oracle evaluations, without any optimality gap on those cases that both formulations are able to solve. This demonstrates the ability of the NDDP handling problems without relatively complete recourses. It is

worth mentioning that for cases where the NDDP algorithm converges without regularization, the computation time used is usually smaller than it spends on the regularized problem, which can be explained by better numerical conditions of feasibility cuts and their effect on reducing the effective volumes of the state space.

4.2 Hydro-Thermal Power Planning Problem

We next consider the Brazilian interconnected power system described in [10]. By assuming the stagewise independence in the underlying stochastic energy inflow, we formulate the problem below as a DR-MCO recursion (2). Let $\mathcal{K} = \{1, \ldots, K\}$ denote the indices of four regions in the system, and $\mathcal{L} = \cup_{k\in\mathcal{K}}\mathcal{L}_k$ the indices of thermal power plants, where each of the disjoint subsets $\mathcal{L}_k$ is associated with the a region $k \in \mathcal{K}$. We first describe the decision variables in each stage $t \in \mathcal{T}$. Let $n \in \mathcal{N}(t)$ denote the index of a sampled outcome in stage t. We use $l_{n,k}$ to denote the stored energy level, $h_{n,k}$ to denote the hydro power generation, and $s_{n,k}$ to denote the energy spillage, of some region $k \in \mathcal{K}$; and $g_{n,l}$ to denote the thermal power generation for some thermal power plant $l \in \mathcal{L}$. For two different regions $k \neq k' \in \mathcal{K}$, we use $e_{n,k,k'}$ to denote the energy exchange from region k to region k', and $a_{n,k,k'}$ to denote the deficit account for region k in region k'. Suppose $(w_{n,k})_{k\in\mathcal{K}}$ is the energy inflow associated with the sampled outcome $n \in \mathcal{N}(t)$, then the stage t subproblem can be written as

$$\mathcal{Q}_{t-1}(x_{t-1}) \coloneqq \tag{35}$$

$$\max_{p_{t-1}\in\mathcal{P}_{t-1}} \sum_{n\in\mathcal{N}(t)} p_{t-1,n} \cdot \min \quad \sum_{k\in\mathcal{K}} \left(c^s s_{n,k} + \sum_{l\in\mathcal{L}_k} c^g_l g_{n,l} + \sum_{k'\in\mathcal{K}} \left(c^e_{k,k'} e_{n,k,k'} + c^a_{k,k'} a_{n,k,k'} \right) \right) + \mathcal{Q}_t(x_n)$$

$$\begin{aligned}
\text{s.t.} \quad & l_{n,k} + h_{n,k} + s_{n,k} = l_{t-1,k} + w_{n,k}, && \forall\, k \in \mathcal{K}, \\
& h_{n,k} + \sum_{l\in\mathcal{L}_k} g_{n,l} + \sum_{k'\in\mathcal{K}} (a_{n,k,k'} - e_{n,k,k'} + e_{n,k',k}) = d_{t,k}, && \forall\, k \in \mathcal{K}, \\
& l_{n,k} \in [0, B^l_k], && \forall\, k \in \mathcal{K}, \\
& h_{n,k} \in [0, B^h_k], && \forall\, k \in \mathcal{K}, \\
& g_{n,l} \in [B^{g,-}_l, B^{g,+}_l], && \forall\, l \in \mathcal{L}, \\
& a_{n,k,k'} \in [0, B^a_{k,k'}], && \forall\, k, k' \in \mathcal{K}, \\
& e_{n,k,k'} \in [0, B^e_{k,k'}], && \forall\, k, k' \in \mathcal{K}.
\end{aligned}$$

Here in the formulation, c^s denotes the unit penalty on energy spillage, c^g_l the unit cost of thermal power generation of plant l, $c^e_{k,k'}$ the unit cost of power exchange from region k to region k', $c^a_{k,k'}$ the unit cost on the energy deficit account for region k in region k', $d_{t,k}$ the deterministic power demand in stage t and region k, B^l_k the bound on the storage level in region k, B^h_k the bound on hydro power generation in region k, $B^{g,-}_l, B^{g,+}_l$ the lower and upper bounds of thermal power generation in plant l, $B^a_{k,k'}$ the bound on the deficit account for region k in region k', and $B^e_{k,k'}$ the bound on the energy exchange from region k to

region k'. The first constraint in (35) characterizes the change of energy storage levels in each region k, the second constraint imposes the power generation-demand balance for each region k, and the rest are bounds on the decision variables. The state variables x_t (resp. x_n) are the energy storage levels $(l_{t,k})_{k\in\mathcal{K}}$ (resp. $(l_{n,k})_{k\in\mathcal{K}}$), while the internal variables consist of all the rest of decision variables. The initial state x_0 is given by data.

The energy inflow outcomes are sampled from multivariate lognormal distributions that are interstage independent. Then the distributional uncertainty set is constructed using Wasserstein metric to reduce the effect of overtraining with the sampled outcome, according to [12]. To be precise, suppose $\hat{p}_{t-1} \in \Delta^N$ is an empirical distribution of outcomes $l_n \coloneqq (l_{n,k})_{k\in\mathcal{K}}$ for $n \in \mathcal{N}(t)$, where $N = \left|\mathcal{N}(t)\right|$ and often $\hat{p}_{t-1} = (1/N, \ldots, 1/N)$. Then, the distributional uncertainty set $\mathcal{P}_{t-1}$ is described by

$$\mathcal{P}_{t-1} \coloneqq \left\{ p_{t-1} \in \Delta^N : \rho(p_{t-1}, \hat{p}_{t-1}) \le \sigma \right\}, \tag{36}$$

for some radius $\sigma \ge 0$, where the Wasserstein metric ρ for finitely supported distributions is defined by

$$\begin{aligned} \rho(p_{t-1}, \hat{p}_{t-1}) \coloneqq \min_{u_{m,n}\ge 0} \quad & \sum_{m,n\in\mathcal{N}(t)} \|l_m - l_n\|\, u_{m,n} \\ \text{s.t.} \quad & \sum_{n\in\mathcal{N}(t)} u_{m,n} = p_{t-1,m}, \quad \forall\, m \in \mathcal{N}(t), \\ & \sum_{m\in\mathcal{N}(t)} u_{m,n} = \hat{p}_{t-1,n}, \quad \forall\, n \in \mathcal{N}(t). \end{aligned} \tag{37}$$

Note when the radius $\sigma = 0$, the distributional uncertainty set $\mathcal{P}_{t-1}$ becomes a singleton. In our numerical tests, we choose the radius to be relative to the total distances, i.e., $\sigma = \beta \cdot \sum_{m,n\in\mathcal{N}(t)} \|l_m - l_n\|$ for some $\beta \ge 0$. At the same time, we use uniform regularization factors for the tests, i.e., $M_t = M > 0$ for all $t \in \mathcal{T}$. When the relative optimality gap α is smaller than the threshold 5%, we check whether all the active cuts in the recent iterations are strictly smaller than the regularization factor. If they are, then the algorithm is terminated, and otherwise the regularization factor M is increased by a factor of $\sqrt{10} \approx 3.1623$ with all the over-approximations reset to $\underline{\mathcal{Q}}_t^i(x) \leftarrow +\infty$, $t \in \mathcal{T}$. Five scenarios are sampled independently in each stage for the nominal problem $N = 5$ before the distributional robust counterpart is constructed by (36). for the 24-stage problem that we consider, the samples already give a total $5^{24} \approx 5.9 \times 10^{16}$ scenario paths, which is practically impossible to solve via an extensive robust formulation. We have then obtained the following results (Table 2) using our Seq-DDP algorithm within a time limit of 5 hours.

In Table 2, the lower bound (LB), the upper bound (UB) at termination, the computation time (Med. Time) and the number of subproblem oracles (Med. `#Eval`) shown are the median of the five test cases. The logarithmic regularization factors $\log_{10}(M)$ listed in the table correspond to the initial regularization factors. We see that for different choices of the relative radii β, the median computation time and number of subproblem oracle evaluations are usually smaller when $\log_{10}(M) = 3$, without compromising the quality of upper and

β	$\log_{10}(M)$	LB ($\cdot 10^7$)	UB ($\cdot 10^7$)	Med. Time (s)	Med. `#Eval`
0.00	2	4.63	4.87	4411.02	35314
	3	4.62	4.86	3635.46	15012
	4	4.60	4.84	5947.57	18377
	5	4.60	4.84	7918.13	17795
0.02	2	4.88	5.14	3676.19	32111
	3	4.84	5.09	2642.95	14663
	4	4.85	5.10	5086.24	16480
	5	4.84	5.10	6333.10	16605
0.04	2	5.11	5.38	2976.62	28469
	3	5.11	5.38	4443.41	23952
	4	5.08	5.35	4757.78	15897
	5	5.09	5.35	5475.20	15465
0.06	2	5.37	5.66	2988.33	29978
	3	5.34	5.62	2000.84	12999
	4	5.33	5.61	3434.14	14401
	5	5.33	5.61	4717.09	14578
0.08	2	5.61	5.90	2642.00	28871
	3	5.59	5.88	1889.54	12243
	4	5.57	5.86	2645.60	13432
	5	5.57	5.86	3338.21	13653
0.10	2	5.85	6.16	2551.69	29052
	3	5.82	6.12	1260.03	10404
	4	5.81	6.11	1713.84	12433
	5	5.81	6.12	2677.05	12660

Table 2: Numerical tests of hydro-thermal power planning problem with different uncertainty radii

lower bounds. Moreover, for $\beta \leq 0.08$, the median computation times for $\log_{10}(M) = 2$ are still smaller than those of $\log_{10}(M) = 4$ or 5, despite the larger number of subproblem oracle evaluations. This can be explained by the better numerical conditions for the smaller regularization factors (the cuts have smaller Lipschitz constants), leading to shorter subproblem oracle evaluations times (cf. Algorithm 1). We thus conclude that the regularization technique could lead to smaller number of subproblem oracle evaluations, as well as shorter computation time for a given DR-MCO problem.

5 Concluding Remarks

In this work, we proposed a new class of algorithms that generalize and strengthen DDP algorithms to solve a broad class of DR-MCO problems. The new algorithms use regularization to effectively control the growth of Lipschitz constants in the approximation and to handle problems without relatively complete recourse. We provide a thorough complexity analysis of the new algorithms, proving both upper complexity bounds and a matching lower bound, which reveal, in a precise way, the dependence of the complexity of the DDP-type algorithms on the number of stages, the dimension of the decision space, and various regularity characteristics of DR-MCO. This is the first complexity analysis of DDP-type algorithms in such a general setting, and we believe it provides key insights for further developing efficient computational tools for the very many applications of sequential decision making under uncertainty. We also provide numerical examples to show the capability of the DDP-type algorithms method to solve problems without relatively complete recourse, and reduction in computation time and number of subproblem oracle evaluations, due to the regularization technique.

References

[1] S Ahmed, F G Cabral, and B F Paulo da Costa. Stochastic lipschitz dynamic programming, 2019.

[2] E J Anderson and A B Philpott. Improving sample average approximation using distributional robustness. 2019.

[3] R Baucke, A Downward, and G Zakeri. A deterministic algorithm for solving multistage stochastic programming problems. *Optimization Online*, 2017.

[4] R Baucke, A Downward, and G Zakeri. A deterministic algorithm for solving stochastic minimax dynamic programmes. *Optimization Online*, 2018.

[5] A Ben-Tal, L El Ghaoui, and A Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[6] D Bertsimas and A Georghiou. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research*, 63(3):610–627, 2015.

[7] John R. Birge. Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs. *Operations Research*, 33(5):989–1007, October 1985.

[8] Z L Chen and W B Powell. Convergent Cutting-Plane and Partial-Sampling Algorithm for Multistage Stochastic Linear Programs with Recourse. *Journal of Optimization Theory and Applications*, 102(3):497–524, September 1999.

[9] Vitor L. de Matos, Andy B. Philpott, and Erlon C. Finardi. Improving the performance of Stochastic Dual Dynamic Programming. *Journal of Computational and Applied Mathematics*, 290:196–208, December 2015.

[10] L Ding, S Ahmed, and A Shapiro. A python package for multi-stage stochastic programming. *Optimization Online*, pages 1–41, 2019.

[11] I Dunning, J Huchette, and M Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[12] D Duque and D P Morton. Distributionally robust stochastic dual dynamic programming. *Optimization Online*, 2019.

[13] Martin Dyer and Leen Stougie. Computational complexity of stochastic programming problems. *Mathematical Programming*, 106(3):423–432, May 2006.

[14] B.C. Flach, L.A. Barroso, and M.V.F. Pereira. Long-term optimal allocation of hydro generation for a price-maker company in a competitive market: Latest developments and a stochastic dual dynamic programming approach. *IET Generation, Transmission & Distribution*, 4(2):299, 2010.

[15] A Georghiou, A Tsoukalas, and W Wiesemann. Robust dual dynamic programming. *Operations Research*, 67(3):813–830, 2019.

[16] P. Girardeau, V. Leclere, and A. B. Philpott. On the Convergence of Decomposition Methods for Multistage Stochastic Convex Programs. *Mathematics of Operations Research*, 40(1):130–145, February 2015.

[17] A Grothey, S Leyffer, and KIM McKinnon. A note on feasibility in benders decomposition. *Numerical Analysis Report NA/188, Dundee University*, 1999.

[18] M J Hadjiyiannis, P J Goulart, and D Kuhn. A scenario approach for estimating the suboptimality of linear decision rules in two-stage robust optimization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7386–7391. IEEE, 2011.

[19] J Huang, K Zhou, and Y Guan. A Study of Distributionally Robust Multistage Stochastic Optimization. *arXiv:1708.07930 [math]*, August 2017.

[20] D Kuhn, W Wiesemann, and A Georghiou. Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130(1):177–209, 2011.

[21] Guanghui Lan. Complexity of Stochastic Dual Dynamic Programming. *arXiv:1912.07702*, December 2019.

[22] K. Linowsky and A. B. Philpott. On the Convergence of Sampling-Based Decomposition Algorithms for Multistage Stochastic Programs. *Journal of Optimization Theory and Applications*, 125(2):349–366, May 2005.

[23] R P Liu. On feasibility of sample average approximation solutions to stochastic programming. *arXiv preprint arXiv:1904.00137*, 2019.

[24] M. V. F. Pereira and L. M. V. G. Pinto. Stochastic Optimization of a Multireservoir Hydroelectric System: A Decomposition Approach. *Water Resources Research*, 21(6):779–792, June 1985.

[25] M. V. F. Pereira and L. M. V. G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1-3):359–375, May 1991.

[26] A Philpott, F Wahid, and F Bonnans. MIDAS: a mixed integer dynamic approximation scheme. *Mathematical Programming*, pages 1–32, 2019.

[27] A. B. Philpott, V. L. de Matos, and L. Kapelevich. Distributionally robust SDDP. *Computational Management Science*, 15(3-4):431–454, October 2018.

[28] A.B. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, July 2008.

[29] A Shapiro. On complexity of multistage stochastic programs. *OR Letters*, 34(1):1–8, January 2006.

[30] A Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, February 2011.

[31] A Shapiro, D Dentcheva, and A Ruszczyński. *Lectures on stochastic programming*. SIAM, 2009.

[32] A Shapiro and L Ding. Stationary multistage programs. *Optimization Online*, 2019.

[33] A Shapiro, W Tekaya, J P da Costa, and M P Soares. Risk neutral and risk averse Stochastic Dual Dynamic Programming method. *European Journal of Operational Research*, 224(2):375–391, January 2013.

[34] S Zhang and X A Sun. Stochastic dual dynamic programming for multistage stochastic mixed-integer nonlinear optimization. *arXiv preprint arXiv:1912.13278*, 2019.

[35] J Zou, S Ahmed, and X A Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175(1-2):461–502, May 2019.

[36] Jikai Zou, Shabbir Ahmed, and Xu Andy Sun. Multistage Stochastic Unit Commitment Using Stochastic Dual Dynamic Integer Programming. *IEEE Transactions on Power Systems*, 34(3):1814–1823, May 2019.