



Complexity of stochastic dual dynamic programming

Guanghui Lan¹

Received: 22 December 2019 / Accepted: 8 September 2020

© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2020

Abstract

Stochastic dual dynamic programming is a cutting plane type algorithm for multi-stage stochastic optimization originated about 30 years ago. In spite of its popularity in practice, there does not exist any analysis on the convergence rates of this method. In this paper, we first establish the number of iterations, i.e., iteration complexity, required by a basic dual dynamic programming method for solving single-scenario multi-stage optimization problems, by introducing novel mathematical tools including the saturation of search points. We then refine these basic tools and establish the iteration complexity for an explorative dual dynamic programming method proposed herein and the classic stochastic dual dynamic programming method for solving more general multi-stage stochastic optimization problems under the standard stage-wise independence assumption. Our results indicate that the complexity of these methods mildly increases with the number of stages T , in fact linearly dependent on T for discounted problems. Therefore, they are efficient for strategic decision making which involves a large number of stages, but with a relatively small number of decision variables in each stage. Without explicitly discretizing the state and action spaces, these methods might also be pertinent to the related reinforcement learning and stochastic control areas.

Mathematics Subject Classification 90C25 · 90C06 · 90C22 · 49M37 · 93A14 · 90C15

Dedicated to Professor Alexander Shapiro on the occasion of his 70th birthday for his profound contributions to stochastic optimization.

This research was partially supported by the NSF grant 1953199 and NIFA Grant 2020-67021-31526.

✉ Guanghui Lan
george.lan@isye.gatech.edu

¹ H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

1 Introduction

In this paper, we are interested in solving the following stochastic dynamic optimization problem

$$\begin{aligned} \min_{x_1 \in X_1} & H_1(x_1, \mathbf{c}_1) + \lambda \mathbb{E} \left[\min_{x_2 \in X_2(x_1)} H_2(x_2, \mathbf{c}_2) \right. \\ & \left. + \lambda \mathbb{E} \left[\cdots + \lambda \mathbb{E} \left[\min_{x_T \in X_T(x_{T-1})} H_T(x_T, \mathbf{c}_T) \right] \right] \right], \end{aligned} \quad (1.1)$$

with feasible sets X_t given by

$$X_1 := \{x \in \bar{X}_1 \subseteq \mathbb{R}^{n_1} : \mathbf{A}_1 x_1 = \mathbf{b}_1, \Phi_1(x_1, \mathbf{p}_1) \leq 0\}, \quad (1.2)$$

$$\begin{aligned} X_t(x_{t-1}) & \equiv X_t(x_{t-1}, \xi_t) \\ & := \{x \in \bar{X}_t \subseteq \mathbb{R}^{n_t} : \mathbf{A}_t x = \mathbf{B}_t x_{t-1} + \mathbf{b}_t, \Phi_t(x, \mathbf{p}_t) \leq \mathbf{Q}_t x_{t-1}\}. \end{aligned} \quad (1.3)$$

Here T denotes the number of stages, $H_t(\cdot, \mathbf{c}_t)$ are closed convex objective functions, $\bar{X}_t \subset \mathbb{R}^{n_t}$ are closed convex sets, $\lambda \in (0, 1]$ denotes the discounting factor, $\mathbf{A}_t : \mathbb{R}^{n_t} \rightarrow \mathbb{R}^{m_t}$, $\mathbf{B}_t : \mathbb{R}^{n_{t-1}} \rightarrow \mathbb{R}^{m_t}$, and $\mathbf{Q}_t : \mathbb{R}^{n_{t-1}} \rightarrow \mathbb{R}^{p_t}$ are linear mappings, and $\Phi_{t,i}(\cdot, \mathbf{p}_t) : \mathbb{R}^{n_t} \rightarrow \mathbb{R}$, $i = 1, \dots, p_t$ are closed convex constraint functions. Moreover, $\xi_1 := (\mathbf{A}_1, \mathbf{b}_1, \mathbf{B}_1, \mathbf{p}_1, \mathbf{c}_1)$ is a given deterministic vector, and $\xi_t := (\mathbf{A}_t, \mathbf{b}_t, \mathbf{B}_t, \mathbf{Q}_t, \mathbf{p}_t, \mathbf{c}_t)$, $t = 2, \dots, T$, are the random vectors at stage t . In particular, if H_t are affine, X_t are polyhedral and Φ_t do not exist, then problem (1.1) reduces to the well-known multi-stage stochastic linear programming problem (see, e.g., [5, 28]). The incorporation of the nonlinear (but convex) objective functions H_t and constraints Φ_t allows us to model a much wider class of problems.

In spite of its wide applicability, multi-stage stochastic optimization remains highly challenging to solve. As shown by Nemirovski and Shapiro [29] and Shapiro [26], the number of scenarios of ξ_t , $t = 2, \dots, T$, required to solve problem (1.1) has to increase exponentially with T . In particular, if the number of stages $T = 3$, the total number of samples (a.k.a. scenarios) should be of order $\mathcal{O}(1/\epsilon^4)$ in general. There exist many algorithms for solving multi-stage stochastic optimization problems (e.g., [11, 21, 24]), but quite often without guarantees provided on their rate of convergence. More recently, Lan and Zhou [17] developed a dynamic stochastic approximation method for multi-stage stochastic optimization by generalizing stochastic gradient descent methods, and show that this algorithm can achieve this optimal sampling and iteration complexity bound for solving general multi-stage stochastic optimization problems with $T = 3$. The complexity of this method depends mildly on the problem dimensions, but increases exponentially with respect to T . As a result, this type of method is suggested for solving some operational decision-making problems, which involve a large number of decision variables but only a small number of stages.

In practice, we often encounter strategic decision making problems which span a long horizon and thus require a large number of stages T . In this situation, a crucial simplification that has been explored to solve problem (1.1) more efficiently is to assume the stage-wise independence. In other words, we make the assumption that

the random variables ξ_t , $t = 2, \dots, T$, are mutually independent of each other. Under this assumption, we can write problem (1.1) equivalently as

$$\min_{x_1 \in X_1} \{H_1(x_1, \mathbf{c}_1) + \lambda V_2(x_1)\}, \quad (1.4)$$

where the value functions V_t , $t = 2, \dots, T$, are recursively defined by

$$\begin{aligned} V_t(x_{t-1}) &:= \mathbb{E}[\mathcal{V}_t(x_{t-1}, \xi_t)], \\ \mathcal{V}_t(x_{t-1}, \xi_t) &:= \min_{x_t \in X_t(x_{t-1})} \{H_t(x_t, \mathbf{c}_t) + \lambda V_{t+1}(x_t)\}, \end{aligned} \quad (1.5)$$

and

$$V_{T+1}(x_T) = 0. \quad (1.6)$$

Furthermore, as pointed out by Shapiro [27], one can generate a relatively small (i.e., N_t) number of samples for each ξ_t and define the so-called sample average approximation (SAA) problem by replacing the expectation in (1.5) with the average over the generated samples (see Sect. 4 for more details).

Under the aforementioned stage-wise independence assumption, a widely-used method for solving the SAA problem is the stochastic dual dynamic programming (SDDP) algorithm. SDDP is an approximate cutting plane method, first presented by Birge [4] and Pereira and Pinto [21] and later studied by Shapiro [27], Philpott et. al. [22], Donohue and Birge [6], Hindsberger [12], Kozmík and Morton [14], Guigues [10] and Zou et. al. [30], among many others. SDDP has been applied to solve problems arising from many different fields such as hydro-thermal planning [9, 30] and biochemical process control [2]. Each iteration of this algorithm contains two phases. In the forward phase, feasible solutions at each stage will be generated starting from the first stage based on the cutting plane models for the value functions built in the previous iteration. Then in the backward phase, the cutting plane models for the value functions of each stage will be updated starting from the last stage. While the cost per iteration of the SDDP method only linearly depends on the number of stages, it remains unknown what is the number of iterations required by the SDDP method to achieve a certain accurate solution of problem (1.4). Existing proofs of convergence of SDDP are based on the assumption that the procedure passes through every possible scenario many times [8, 19, 27]. Of course when the number of scenarios, although finite, is astronomically large this is not very realistic. In addition, such analysis does not reveal the dependence of the efficiency of SDDP on various parameters, e.g., number of stages, target accuracy, Lipschitz constants, and diameter of feasible sets etc.

It is well-known that when the number of stages $T = 2$, SDDP reduces to the classic Kelley's cutting plane method [13]. As shown in Nesterov [20], the number of iterations required by Kelley's cutting plane method could depend exponentially on the dimension of the problem even for a static optimization problem inevitably. Therefore, this type of method is not recommended for solving large-scale optimization problems. However, it turns out that the global cutting plane models are critically important for multi-stage optimization especially if the number of stages is large and one does not know the structure of optimal policies. In these cases we need to understand the

efficiency of these cutting plane methods in order to identify not only problem classes amenable for these techniques, but also possibly to inspire new ideas to solve these problems more efficiently.

This paper intends to close the aforementioned gap in our understanding about cutting plane methods for multi-stage stochastic optimization. Our main contributions mainly exist in the following several aspects. Firstly, we start with a dual dynamic programming (DDP) method for solving dynamic convex optimization problem with a single scenario. This simplification allows us to build a few essential mathematical notions and tools for the analysis of cutting plane methods. More specifically, we introduce the notion of saturated and distinguishable search points. Using this notion, we show that each iteration of DDP will either find a new saturated and distinguishable search point, or compute an approximate solution for the original problem. As a consequence, we establish the total number of iterations required by the DDP method for solving the single-scenario problem. More specifically, we show that the iteration complexity of DDP only mildly increases w.r.t. the number of stages T , in fact linearly dependent on T for many problems, especially those with a discounting factor $\lambda < 1$. The dependence of DDP on other problem parameters has also been thoroughly studied. We also demonstrate that one can terminate DDP based on some easily computable upper and lower bounds on the optimal value.

Secondly, motivated by the analysis of the DDP method, we propose a new explorative dual dynamic programming (EDDP) for solving the SAA problem of multi-stage stochastic optimization in (1.4). When solving the SAA problem, we have to choose one out of N_t possible feasible solutions in the forward phase, and each one of them corresponds to a random realization of ξ_t . In EDDP, we choose a feasible solution in an aggressive manner by selecting the most distinguishable search point among the saturated ones in each stage. As a result, we show that the number of iterations required by EDDP for solving the SAA problem is the same as that of DDP for solving the single-scenario problem. However, to implement EDDP we need to maintain the set of saturated search points explicitly.

Thirdly, we show that the SDDP method can be viewed as a randomized version of the EDDP algorithm by choosing the aforementioned feasible solution at each stage t randomly from the N_t possible selections. Since this algorithm is stochastic, we establish the expected number of iterations required by SDDP to compute an approximate feasible policy for solving the SAA problem. In particular the iteration complexity of SDDP is worse than that of DDP and EDDP by a factor of $\bar{N} := \max\{N_2, \dots, N_T\}$, but still mildly increases w.r.t. T . Moreover, we show that the probability of having large deviation from this expected iteration complexity decays exponentially fast. In addition, we establish the convergence of the gap between a stochastic upper bound and lower bound on the optimal value, and show how we can possibly use these bounds to terminate the algorithm.

To the best of our knowledge, all the aforementioned complexity results, as well as the analysis techniques, are new for cutting plane methods for multi-stage stochastic optimization.

This paper is organized as follows. In Sect. 2, we present some preliminary results on the basic cutting plane methods for solving static convex optimization problems. In Sect. 3, we present the DDP method for single-scenario problems and establish its

convergence properties. Section 4 is devoted to the EDDP method for solving the SAA problem for multi-stage stochastic optimization. In Sect. 5, we establish the complexity of the SDDP method. Finally, some concluding remarks are made in Sect. 6.

2 Preliminary: Kelley's cutting plane methods

In this section, we briefly review the basic cutting plane method and establish its complexity bound. Consider the convex programming problem of

$$\min_{x \in X} f(x), \quad (2.1)$$

where $X \subseteq \mathbb{R}^n$ is a convex compact set and $f : X \rightarrow \mathbb{R}$ is a sub-differentiable convex function. Moreover, we assume that f is Lipschitz continuous s.t.

$$|f(x) - f(y)| \leq M\|x - y\|, \forall x, y \in X. \quad (2.2)$$

Algorithm 1 formally describes Kelley's cutting plane method for solving (2.1). The essential construct in this algorithm is the cutting plane model $\underline{f}(x)$, which always underestimates $f(x)$ for any $x \in X$. Given the current search point x_k , this method first updates the model function \underline{f} and then minimizes it to compute the new search point x_{k+1} . It terminates if the gap between the upper bound (ub_k) and lower bound (lb_k) falls within the prescribed target accuracy ϵ . As a result, an ϵ -solution $\bar{x} \in X$ s.t. $f(\bar{x}) - f(x^*) \leq \epsilon$ will be found whenever the algorithm stops.

Algorithm 1 Basic cutting plane method

Input: initial points x_1 and target accuracy ϵ .
 Set $\underline{f}_0(x) = -\infty$ and $ub_0 = +\infty$.
for $k = 1, 2, \dots$, **do**
 Set $\underline{f}_k(x) = \max\{\underline{f}_{k-1}(x), f(x_k) + \langle f'(x_k), x - x_k \rangle\}$.
 Set $x_{k+1} \in \operatorname{Argmin}_{x \in X} \underline{f}(x)$.
 Set $lb_k = \underline{f}(x_{k+1})$ and $ub_k = \min\{ub_{k-1}, f(x_{k+1})\}$.
if $ub_k - lb_k \leq \epsilon$ **then**
 terminate.
end if
end for

We establish the complexity, i.e., the number of iterations required to have a gap lower than ϵ , of the cutting plane method in Proposition 1.

Proposition 1 *Unless Algorithm 1 stops, we have $\|x_{k+1} - x_i\| \geq \epsilon/M$ for any $i = 1, \dots, k$. Moreover, suppose that the norm $\|\cdot\|$ in (2.2) is given by the l_∞ norm and $X \subset \mathbb{R}^n$ is contained in a box with side length bounded by l . Then the complexity of the basic cutting plane method can be bounded by*

$$\left(\frac{lM}{\epsilon} + 1\right)^n. \quad (2.3)$$

Proof Note that $\underline{f}_k(x) = \max_{i=1,\dots,k} f(x_i) + \langle f'(x_i), x - x_i \rangle$ is Lipschitz continuous with constant M . Moreover, we have $\underline{f}_k(x) \leq f(x)$ for any $x \in X$ and $f(x_i) = \underline{f}_k(x_i)$ for any $i = 1, \dots, k+1$. Hence,

$$\underline{f}_k(x_{k+1}) = \min_{x \in X} \underline{f}_k(x) \leq \min_{x \in X} f(x) = f^*.$$

Using this observation, we have

$$\text{ub}_k - \text{lb}_k \leq f(x_i) - \text{lb}_k = \underline{f}_k(x_i) - \text{lb}_k = \underline{f}_k(x_i) - \underline{f}_k(x_{k+1}) \leq M \|x_i - x_{k+1}\|.$$

Since $\text{ub}_k - \text{lb}_k > \epsilon$, we must have $\|x_i - x_{k+1}\| > \epsilon/M$. (2.3) then follows immediately from this observation. \square

Even though the complexity bound (2.3) of the cutting plane method has not been explicitly established before, construction of this proof was used in Ruszczyński [25]. Moreover, as pointed out in [20] the exponential dependence of such complexity bound on the dimension n does not seem to be improvable in general. It is worth noting that the cutting plane algorithm does not explicitly depend on the selection of the norm even though the bound in (2.3) is obtained under the assumption that X sits inside an l_∞ box.

3 Dual dynamic programming for single-scenario problems

In this section, we focus on a dynamic version of the cutting plane method applied to solve a class of deterministic dynamic convex optimization problems, i.e., multi-stage optimization problems with a single scenario. This dual dynamic programming (DDP) method, which can be viewed as SDDP with one scenario, will serve as a starting point for studying the more general dual dynamic programming methods in later two sections. Moreover, this method may inspire some interests in its own right.

More specifically, we consider the following dynamic convex programming

$$f^* := \min_{x_1 \in X_1} \{f_1(x_1) := h_1(x_1) + \lambda v_2(x_1)\}, \quad (3.1)$$

where the value functions $v_t(\cdot)$, $t = 2, \dots, T+1$, are defined recursively by

$$v_t(x_{t-1}) := \min_{x_t \in X_t(x_{t-1})} \{f_t(x_t) := h_t(x_t) + \lambda v_{t+1}(x_t)\}, \quad (3.2)$$

$$v_{T+1}(x_T) \equiv 0, \quad (3.3)$$

with convex feasible sets $X_t(x_{t-1})$ given by

$$X_t(x_{t-1}) := \{x \in \bar{X}_t \subseteq \mathbb{R}^{n_t} : A_t x = B_t x_{t-1} + b_t, \phi_t(x) \leq Q_t x_{t-1}\}. \quad (3.4)$$

Similarly to problem (1.1), here $\bar{X}_t \subset \mathbb{R}^{n_t}$ are closed convex sets independent of x_{t-1} , $\lambda \in (0, 1]$ denotes the discounting factor, $A_t : \mathbb{R}^{n_t} \rightarrow \mathbb{R}^{m_t}$, $B_t : \mathbb{R}^{n_{t-1}} \rightarrow \mathbb{R}^{m_t}$, and

$Q_t : \mathbb{R}^{n_{t-1}} \rightarrow \mathbb{R}^{p_t}$ are linear mappings, and $h_t : \bar{X}_t \rightarrow \mathbb{R}$ and $\phi_{t,i} : \bar{X}_t \rightarrow \mathbb{R}$, $i = 1, \dots, p_t$, are closed convex functions. Thus, we can view problem (3.1) as a single-scenario multi-stage optimization problem in the form of (1.1), by assuming $\xi_t = (A_t, b_t, B_t, Q_t, p_t, c_t)$ to be deterministic, and setting $h_t(\cdot) = H_t(\cdot, c_t)$ and $\phi_t(\cdot) = \Phi_t(\cdot, p_t)$.

Throughout this section, we denote \mathcal{X}_t the effective feasible region of each period t defined recursively by

$$\mathcal{X}_t := \begin{cases} X_1, & t = 1, \\ \cup_{x \in \mathcal{X}_{t-1}} X_t(x), & t \geq 2. \end{cases} \quad (3.5)$$

Observe that \mathcal{X}_t is not necessarily convex and its convex hull is denoted by $\text{Conv}(\mathcal{X}_t)$. Moreover, letting $\text{Aff}(\mathcal{X}_t)$ be the affine hull of \mathcal{X}_t and $\mathcal{B}_t(\epsilon) := \{y \in \text{Aff}(\mathcal{X}_t) : \|y\| \leq \epsilon\}$, we use

$$\mathcal{X}_t(\epsilon) := \mathcal{X}_t + \mathcal{B}_t(\epsilon)$$

to denote \mathcal{X}_t together with its surrounding neighborhood.

In order to develop a cutting plane algorithm for solving problem (3.1), we need to make a few assumptions and discuss a few quantities that characterize the problem.

Assumption 1 For any $t \geq 1$, there exists $D_t \geq 0$ s.t.

$$\|x_t - x'_t\| \leq D_t, \quad \forall x_t, x'_t \in \mathcal{X}_t, \quad \forall t \geq 1. \quad (3.6)$$

The quantity D_t provides a bound on the “diameter” of the effective feasible region \mathcal{X}_t . Clearly, Assumption 1 holds if the convex sets \bar{X}_t are compact, since by definition we have $\mathcal{X}_t \subseteq \text{Conv}(\mathcal{X}_t) \subseteq \bar{X}_t$, $\forall t \geq 1$.

Assumption 2 For any $t \geq 1$, there exists $\bar{\epsilon}_t \in (0, +\infty)$ s.t.

$$h_t(x) < +\infty, \quad \forall x \in \mathcal{X}_t(\bar{\epsilon}_t) \quad \text{and} \quad \text{rint}(X_{t+1}(x)) \neq \emptyset, \quad \forall x \in \mathcal{X}_t(\bar{\epsilon}_t), \quad (3.7)$$

where $\text{rint}(\cdot)$ denotes the relative interior of a convex set.

Assumption 2 describes certain regularity conditions of problem (3.1). Specifically, the two conditions in (3.7) imply that h_t and v_{t+1} are finitely valued in $\mathcal{X}_t(\bar{\epsilon}_t)$. The second relation in (3.7) also implies the Slater condition of the feasible sets in (3.4) and thus the existence of optimal dual solutions to define the cutting plane models for problem (3.1). Here the relative interior is required due to the nonlinearity of the constraint functions in (3.4) and we can replace $\text{rint}(X_{t+1}(x))$ with $X_{t+1}(x)$ if the latter is polyhedral. Conditions of these types have been referred to as extended relatively complete recourse, which is less stringent than imposing complete recourse with $\bar{\epsilon} = +\infty$ in the second relation in (3.7) (see [8]).

In view of Assumption 2, the objective functions f_t , as given by the summation of h_t and λv_{t+1} , must be finitely valued in $\mathcal{X}_t(\bar{\epsilon}_t)$. In addition, by Assumption 1 the set \mathcal{X}_t is bounded. Hence the convex functions f_t must be Lipschitz continuous over \mathcal{X}_t (see, e.g., Section 2.2.4 of [15]). We explicitly state the Lipschitz constants of f_t below since they will be used in the convergence analysis our algorithm.

Assumption 3 For any $t \geq 1$, there exists $M_t \geq 0$ s.t.

$$|f_t(x_t) - f_t(x'_t)| \leq M_t \|x_t - x'_t\|, \quad \forall x_t, x'_t \in \mathcal{X}_t. \quad (3.8)$$

We are now ready to describe a dual dynamic programming method for solving problem (3.1) (see Algorithm 2). For notational convenience, we assume that $X_1(x_0^k) \equiv X_1$ for any iteration $k \geq 1$.

Algorithm 2 Dual dynamic programming (DDP) for single-scenario problems

1: Set $\underline{v}_t^0(x) = -\infty$, $t = 2, \dots, T$, $\underline{v}_{T+1}^0 = 0$, and $\text{ub}_t^0 = +\infty$, $t = 1, \dots, T$.
2: **for** $k = 1, 2, \dots$, **do**

3: **for** $t = 1, 2, \dots, T$ **do** ▷ Forward phase.

$$x_t^k \in \text{Argmin}_{x_t} \left\{ \underline{f}_t^{k-1}(x) := h_t(x) + \lambda \underline{v}_{t+1}^{k-1}(x) : x \in X_t(x_{t-1}^k) \right\}. \quad (3.9)$$

4: **end for**
5: Set $\text{ub}_1^k = \min\{\text{ub}_1^{k-1}, \sum_{t=1}^T \lambda^{t-1} h_t(x_t^k)\}.$
6:
7: Set $\underline{v}_{T+1}^k = 0$. ▷ Backward phase.
8: **for** $t = T, T-1, \dots, 2$ **do**

$$\tilde{v}_t^k(x_{t-1}^k) = \min \left\{ \underline{f}_t^k(x) := h_t(x) + \lambda \underline{v}_{t+1}^k(x) : x \in X_t(x_{t-1}^k) \right\}. \quad (3.10)$$

$(\tilde{v}_t^k)'(x_{t-1}^k) = [B_t, Q_t]y_t^k$, where y_t^k is the optimal dual multiplier of (3.10).

$$\underline{v}_t^k(x) = \max \left\{ \underline{v}_t^{k-1}(x), \tilde{v}_t^k(x_{t-1}^k) + \langle (\tilde{v}_t^k)'(x_{t-1}^k), x - x_{t-1}^k \rangle \right\}. \quad (3.11)$$

9: **end for**

10: **end for**

We now make a few observations about the above DDP method. Firstly, in the forward phase our goal is to compute a new policy $(x_1^k, x_2^k, \dots, x_T^k)$ sequentially starting from x_1^k for the first stage. In this phase we utilize the cutting plane model $\underline{v}_{t+1}^{k-1}(\cdot)$ as a surrogate for the value function $v_{t+1}(\cdot)$ in order to approximate the objective function $f_t(\cdot)$ at stage t , because we do not have a convenient expression for the value function $v_{t+1}(\cdot)$. Since $(x_1^k, x_2^k, \dots, x_T^k)$ is a feasible policy by definition, $\sum_{t=1}^T \lambda^{t-1} h_t(x_t^k)$ gives us an upper bound on the optimal value f^* of problem (3.1), and accordingly, ub_1^k gives us the value associated with the best policy we found so far.

Secondly, given the new generated policy $(x_1^k, x_2^k, \dots, x_T^k)$, our goal in the backward phase is to update the cutting plane models $\underline{v}_t^{k-1}(\cdot)$ to $\underline{v}_t^k(\cdot)$, in order to provide a possibly tighter approximation of $v_t(\cdot)$. More specifically, by Assumption 2, the feasible region of $X_t(x_{t-1}^k)$ of the subproblem in (3.10) has a nonempty relative interior. Hence the function value $\tilde{v}_t^k(x_{t-1}^k)$ and the associated vector $[B_t, Q_t]y_t^k$ are well-defined, and they define a supporting hyperplane for the approximate value function $\tilde{v}_t^k(\cdot)$ defined in (3.10) (after replacing x_{t-1}^k with any $x \in \mathcal{X}_{t-1}(\bar{\epsilon}_{t-1})$). Using all these supporting hyperplanes of \tilde{v}_t^k that have been generated so far, we define a cutting plane

model $\underline{v}_t^k : \mathbb{R}^{n_t} \rightarrow \mathbb{R}$, which underestimates the original value function $v_t(\cdot)$ as shown in the following result.

Lemma 1 *For any $k \geq 1$,*

$$\underline{v}_t^{k-1}(x) \leq \underline{v}_t^k(x) \leq \tilde{v}_t^k(x) \leq v_t(x), \forall x \in \mathcal{X}_{t-1}(\bar{\epsilon}_{t-1}), t = 2, \dots, T, \quad (3.12)$$

$$\underline{f}_t^{k-1}(x) \leq \underline{f}_t^k(x) \leq f_t(x), \forall x \in \mathcal{X}_t(\bar{\epsilon}_t), t = 1, \dots, T. \quad (3.13)$$

Proof First observe that the inequalities in (3.13) follow directly from (3.12) by using the facts that $f_t(x) = h_t(x) + \lambda v_{t+1}(x)$ and $\underline{f}_t^k(x) = h_t(x) + \lambda \underline{v}_{t+1}^k(x)$ due to the definitions of f_t and \underline{f}_t^k in (3.2) and (3.9), respectively. Moreover, the first relation $\underline{v}_t^{k-1}(x) \leq \underline{v}_t^k(x)$ follows directly from (3.11).

Second, we observe that the functions \tilde{v}_t^k and v_t are well-defined over $\mathcal{X}_{t-1}(\bar{\epsilon}_{t-1})$ due to Assumption 2 and will show that the remaining inequalities in (3.12), i.e., $\underline{v}_t^k(x) \leq \tilde{v}_t^k(x) \leq v_t(x), \forall x \in \mathcal{X}_{t-1}(\bar{\epsilon}_{t-1})$, hold by using induction backwards for $t = T, \dots, 1$ at any iteration k . Let us first consider $t = T$. Note that $\underline{v}_{T+1}^k = 0$ and thus by comparing the definitions of $v_T(x)$ and $\tilde{v}_T^k(x)$ in (3.2) and (3.10), we have $\tilde{v}_T^k(x) = v_T(x)$. Moreover, by definition $\tilde{v}_T^k(x_{T-1}^k) + \langle (\tilde{v}_T^k)'(x_{T-1}^k), x - x_{T-1}^k \rangle$ is a supporting hyperplane of $\tilde{v}_T^k(x)$ at x_{T-1}^k . Combining these observations with the definition of $\underline{v}_T^k(x)$, we have

$$\underline{v}_T^k(x) \leq \tilde{v}_T^k(x_{T-1}^k) + \langle (\tilde{v}_T^k)'(x_{T-1}^k), x - x_{T-1}^k \rangle \leq \tilde{v}_T^k(x) = v_T(x). \quad (3.14)$$

Now assume that $\underline{v}_t^k(x) \leq \tilde{v}_t^k(x) \leq v_t(x)$ for some $0 \leq t \leq T$. Using the induction hypothesis of $\underline{v}_t^k(x) \leq v_t(x)$ in the definitions of $v_{t-1}(x)$ and $\tilde{v}_{t-1}^k(x)$ in (3.2) and (3.10), we conclude that $\tilde{v}_{t-1}^k(x) \leq v_{t-1}(x)$. Moreover, by definition $(\tilde{v}_{t-1}^k)'(x_{t-2}^k)$ is a subgradient of $\tilde{v}_{t-1}^k(\cdot)$ at x_{t-2}^k . Combining these relations, we conclude

$$\underline{v}_{t-1}^k(x) = \tilde{v}_{t-1}^k(x_{t-2}^k) + \langle (\tilde{v}_{t-1}^k)'(x_{t-2}^k), x - x_{t-2}^k \rangle \leq \tilde{v}_{t-1}^k(x) \leq v_{t-1}(x). \quad (3.15)$$

□

In order to establish the complexity of Algorithm 2, we need to show that the approximation functions $\underline{f}_t^k(\cdot)$ are Lipschitz continuous on \mathcal{X}_t .

Lemma 2 *For any $t \geq 1$, there exists $\underline{M}_t \geq 0$ s.t.*

$$|\underline{f}_t^k(x_t) - \underline{f}_t^k(x'_t)| \leq \underline{M}_t \|x_t - x'_t\|, \forall x_t, x'_t \in \mathcal{X}_t \forall k \geq 1. \quad (3.16)$$

Proof Note that by Assumption 2, for any $x \in \mathcal{X}_t(\bar{\epsilon})$, the feasible region of $X_{t+1}(x)$ has a nonempty relative interior, hence for any $i = 1, \dots, k$, the function values $\tilde{v}_{t+1}^i(x_t^i)$ and the associated vectors $[B_{t+1}, Q_{t+1}]y_{t+1}^i$ are well-defined. Therefore, the piecewise linear function $\underline{v}_{t+1}^k(x)$ given by

$$\underline{v}_{t+1}^k(x) = \max_{i=1, \dots, k} \tilde{v}_{t+1}^i(x_t^i) + \langle [B_{t+1}, Q_{t+1}]y_{t+1}^i, x - x_t^i \rangle$$

is well-defined and sub-differentiable. This observation, in view of the convexity of h_t and Assumption 2, then implies that $\underline{f}_t^k(x) = h_t(x) + \lambda \underline{v}_t^k(x)$ is sub-differentiable on \mathcal{X}_t . We now provide a bound for the subgradients $(\underline{f}_t^k)'$ on \mathcal{X}_t . Note that for any $x \in \mathcal{X}_t(\bar{\epsilon})$ and $x_0 \in \mathcal{X}_t$, we have

$$\langle (\underline{f}_t^k)'(x_0), x - x_0 \rangle \leq \underline{f}_t^k(x) - \underline{f}_t^k(x_0) \leq f(x) - \underline{f}_t^1(x_0), \quad (3.17)$$

where the last inequality follows from (3.13). Letting $\|\cdot\|_* := \max_{\|x\| \leq 1} \langle \cdot, x \rangle$ denotes the conjugate norm of $\|\cdot\|$ and setting $x = x_0 + \bar{\epsilon}(\underline{f}_t^k)'(x_0)/\|(\underline{f}_t^k)'(x_0)\|_*$, we have

$$\bar{\epsilon} \|(\underline{f}_t^k)'(x_0)\|_* \leq f(x) - \underline{f}_t^1(x_0) \leq \max_{x \in \mathcal{X}(\bar{\epsilon})} f(x) - \min_{x \in \mathcal{X}} \underline{f}_t^1(x),$$

which implies that

$$\|(\underline{f}_t^k)'(x_0)\|_* \leq \frac{1}{\bar{\epsilon}} [\max_{x \in \mathcal{X}_t(\bar{\epsilon})} f(x) - \min_{x \in \mathcal{X}_t} \underline{f}_t^1(x)], \forall x_0 \in \mathcal{X}_t.$$

The result in (3.16) then follows directly from the above inequality, the boundedness of \mathcal{X}_t and hence $\mathcal{X}(\bar{\epsilon})$, and the fact that

$$|\underline{f}_t^k(x_t) - \underline{f}_t^k(x'_t)| \leq \max\{\|(\underline{f}_t^k)'(x_t)\|_*, \|(\underline{f}_t^k)'(x'_t)\|_*\} \|x_t - x'_t\|$$

due to the convexity of \underline{f}_t and the Cauchy Schwarz inequality. \square

We now add some discussions about the Lipschitz continuity of \underline{f}_t^k obtained in Lemma 2. Firstly, it might be interesting to establish some relationship between the Lipschitz constants \underline{M}_t and M_t for \underline{f}_t^k and f_t , respectively. Under certain circumstances we can provide such a relationship. In particular, let us suppose that

$$\underline{f}_t^k(x_0) \leq f_t(x_0) \leq \underline{f}_t^k(x_0) + \bar{\epsilon}. \quad (3.18)$$

It then follows from the above assumption and (3.17) that

$$\langle (\underline{f}_t^k)'(x_0), x - x_0 \rangle \leq f(x) - f(x_0) + \bar{\epsilon}.$$

Setting $x = x_0 + \bar{\epsilon} \underline{f}'(x_0)/\|\underline{f}'(x_0)\|_*$, we conclude

$$\bar{\epsilon} \|(\underline{f}_t^k)'(x_0)\|_* \leq f(x) - f(x_0) + \bar{\epsilon} \leq M \|x - x_0\| + \bar{\epsilon} \leq \bar{\epsilon} M + \bar{\epsilon},$$

which implies that

$$\|\underline{f}'(x_0)\|_* \leq M_t + 1 \text{ and } \underline{M}_t \leq M_t + 1. \quad (3.19)$$

Note however that the above relationship does not necessarily hold for a situation more general than (3.18).

Secondly, while it is relatively easy to understand how the discounting factor λ impacts the Lipschitz constants M_t for the objective functions f_t over different stages, its impact on the Lipschitz constants \underline{M}_t for the approximation functions \underline{f}_t^k is more complicated since we do not know how the Lagrange multipliers y_t^k changes w.r.t. λ . On the other hand, the discounting factor does play a role in compensating the approximation errors accumulated over different stages for the DDP method. Since we cannot quantify precisely such a compensation by simply scaling the Lipschitz constants M_t and \underline{M}_t , we decide to incorporate explicitly the discounting factor λ into our problem formulation, as well as the analysis of our algorithms. We will see that to incorporate λ just makes some calculations, but not the major development of the analysis, more complicated. One can certainly assume that $\lambda = 1$ in order to see the basic idea of our convergence analysis.

In order to establish the complexity of DDP, we need to introduce an important notion as follows.

Definition 1 We say that a search point x_t^k gets ϵ_t -saturated at iteration k if

$$v_{t+1}(x_t^k) - \underline{v}_{t+1}^k(x_t^k) \leq \epsilon_t. \quad (3.20)$$

In view of the above definition and (3.12), for any ϵ_t -saturated point x_t^k we must have

$$\underline{v}_{t+1}^k(x_t^k) \leq v_{t+1}(x_t^k) \leq \underline{v}_{t+1}^k(x_t^k) + \epsilon_t. \quad (3.21)$$

In other words, \underline{v}_{t+1}^k will be a tight approximation of v_{t+1} at x_t^k with error bounded by ϵ_t . By (3.12), we also have $\underline{v}_{t+1}^k(x_t^k) \leq \underline{v}_{t+1}^{k'}(x_t^k)$ for any $k' \geq k$, and hence

$$v_{t+1}(x_t^k) - \underline{v}_{t+1}^{k'}(x_t^k) \leq v_{t+1}(x_t^k) - \underline{v}_{t+1}^k(x_t^k) \leq \epsilon_t.$$

This implies that once a point x_t^k becomes ϵ_t -saturated at the k -th iteration, the functions $\underline{v}_{t+1}^{k'}$ will also be a tight approximation of v_{t+1} at x_t^k with error bounded by ϵ_t for any iteration $k' \geq k$.

Below we describe some basic properties about the saturation of the search points.

Lemma 3 Any search point x_{T-1}^k generated for the $(T-1)$ -th stage must be 0-saturated for any $k \geq 1$.

Proof Note that by (3.12), we have $\underline{v}_T^k(x_{T-1}^k) \leq v(x_{T-1}^k)$. Moreover, by (3.11),

$$\underline{v}_T^k(x_{T-1}^k) \geq \tilde{v}_T^k(x_{T-1}^k) = v(x_{T-1}^k)$$

where the last equality follows from the fact that $v_{T+1}^k = 0$ and the definitions of $v_T(x)$ and $\tilde{v}_T^k(x)$ in (3.2) and (3.10). Therefore we must have $\underline{v}_T^k(x_{T-1}^k) = v(x_{T-1}^k)$, which, in view of (3.20), implies that x_{T-1}^k is 0-saturated. \square

We now state a crucial observation for DDP that relates the saturation of search points across two consecutive stages. More specifically, the following result shows that if one search point x_t^j at stage t has been ϵ_t -saturated at iteration j , and a new search point generated at a later iteration k is close to x_t^j , then a search point in the previous stage $t-1$ will get ϵ_{t-1} -saturated with an appropriately chosen value for ϵ_{t-1} .

Proposition 2 *Suppose that the search point x_t^k generated at the k -th iteration is close enough to x_t^j generated in a previous iteration $1 \leq j < k$, i.e.,*

$$\|x_t^k - x_t^j\| \leq \delta_t \quad (3.22)$$

for some $\delta_t \in [0, +\infty)$. Also assume that the search point x_t^j is ϵ_t -saturated, i.e.,

$$v_{t+1}(x_t^j) - \underline{v}_{t+1}^j(x_t^j) \leq \epsilon_t. \quad (3.23)$$

Then we have

$$\begin{aligned} f_t(x_t^k) - \underline{f}_t^{k-1}(x_t^k) &= \lambda[v_{t+1}(x_t^k) - \underline{v}_{t+1}^{k-1}(x_t^k)] \\ &\leq \epsilon_{t-1} := (M_t + \underline{M}_t)\delta_t + \lambda\epsilon_t. \end{aligned} \quad (3.24)$$

In addition, for any $t \geq 2$, we have

$$v_t(x_{t-1}^k) - \underline{v}_t^k(x_{t-1}^k) \leq \epsilon_{t-1} \quad (3.25)$$

and hence the search point x_{t-1}^k will get ϵ_{t-1} -saturated at iteration k .

Proof By the definitions of f_t and \underline{f}_t^{k-1} in (3.2) and (3.9), we have

$$f_t(x) - \underline{f}_t^{k-1}(x) = \lambda[v_{t+1}(x) - \underline{v}_{t+1}^{k-1}(x)], \forall x \in X_t(x_{t-1}^k)$$

and hence first identity in (3.24) holds. It follows from the definition of x_t^k in (3.9) and the first relation in (3.13) that

$$\begin{aligned} f_t(x_t^k) - \min_{x \in X_t(x_{t-1}^k)} \underline{f}_t^{k-1}(x) &= f_t(x_t^k) - \underline{f}_t^{k-1}(x_t^k) \\ &\leq f_t(x_t^k) - \underline{f}_t^j(x_t^k). \end{aligned} \quad (3.26)$$

Now by (3.8) and (3.16), we have

$$|f_t(x_t^k) - f_t(x_t^j)| \leq M_t \|x_t^k - x_t^j\| \text{ and } |\underline{f}_t^j(x_t^k) - \underline{f}_t^j(x_t^j)| \leq \underline{M}_t \|x_t^k - x_t^j\|.$$

In addition, by (3.23) and the definition f_t and \underline{f}_t^j , we have

$$f_t(x_t^j) - \underline{f}_t^j(x_t^j) = \lambda[v_{t+1}(x_t^j) - \underline{v}_{t+1}^j(x_t^j)] \leq \lambda\epsilon_t.$$

Combining the previous observations and (3.22), we have

$$\begin{aligned} f_t(x_t^k) - \underline{f}_t^{k-1}(x_t^k) &\leq [f_t(x_t^k) - f_t(x_t^j)] + [f_t(x_t^j) - \underline{f}_t^j(x_t^j)] + [\underline{f}_t^j(x_t^j) - \underline{f}_t^j(x_t^k)] \\ &\leq (M_t + \underline{M}_t) \|x_t^k - x_t^j\| + \lambda \epsilon_t \\ &\leq (M_t + \underline{M}_t) \delta_t + \lambda \epsilon_t = \epsilon_{t-1}, \end{aligned} \quad (3.27)$$

where the last equality follows from the definition of ϵ_{t-1} in (3.24). Thus we have shown the inequality in (3.24).

We will now show that the search point x_k^{t-1} in the preceding stage $t-1$ must also be ϵ_{t-1} -saturated at iteration k . Note that x_t^k is a feasible solution for the t -th stage problem and hence that the function value $f_t(x_t^k)$ must be greater than the optimal value $v_t(x_{t-1}^k)$. Using this observation, we have

$$v_t(x_{t-1}^k) - \underline{v}_t^k(x_{t-1}^k) \leq f_t(x_t^k) - \underline{v}_t^k(x_{t-1}^k). \quad (3.28)$$

Moreover, using the definitions of $\tilde{v}_t^k(x_{t-1}^k)$ and $\underline{v}_t^k(x_{t-1}^k)$ in (3.10) and (3.11), the relations in (3.12) and the fact that $\underline{v}_{t+1}^k(x) \geq \underline{v}_{t+1}^{k-1}(x)$ due to (3.13), we have

$$\begin{aligned} \underline{v}_t^k(x_{t-1}^k) &= \max\{\underline{v}_t^{k-1}(x_{t-1}^k), \tilde{v}_t^k(x_{t-1}^k)\} \\ &= \tilde{v}_t^k(x_{t-1}^k) \\ &= \min\left\{\underline{f}_t^k(x) : x \in X_t(x_{t-1}^k)\right\} \\ &\geq \min\left\{\underline{f}_t^{k-1}(x) : x \in X_t(x_{t-1}^k)\right\} \\ &= \underline{f}_t^{k-1}(x_t^k), \end{aligned} \quad (3.29)$$

where the last identity follows from the definition of x_t^k in (3.9). Putting together (3.28) and (3.29), we have

$$\begin{aligned} v_t(x_{t-1}^k) - \underline{v}_t^k(x_{t-1}^k) &\leq f_t(x_t^k) - \underline{f}_t^{k-1}(x_t^k) \\ &\leq \epsilon_{t-1}, \end{aligned} \quad (3.30)$$

where the last inequality follows from (3.27). The above inequality then implies that x_{t-1}^k gets ϵ_{t-1} -saturated at the k -th iteration. \square

Observe that the functions $f_t(\cdot)$ are not directly computable since they depend on the exact value functions $v_{t+1}(\cdot)$. The following result relates the notion of saturation to the gap between a computable upper bound $\sum_{t=1}^T \lambda^{t-1} h_t(x_t^k)$ and the lower bound $\underline{f}_1^{k-1}(x_1^k)$ on the optimal value f^* , under the assumption that the concluding inequality (3.24) obtained in Proposition 2 holds for all the stages, i.e., $\lambda[v_{t+1}(x_t^k) - \underline{v}_{t+1}^{k-1}(x_t^k)] \leq \epsilon_{t-1}$, $\forall t = 1, \dots, T$.

Lemma 4 Suppose that at some iteration $k \geq 1$, we have

$$\lambda[v_{t+1}(x_t^k) - \underline{v}_{t+1}^{k-1}(x_t^k)] \leq \epsilon_{t-1}, \quad (3.31)$$

for any $t = 1, \dots, T$. Then we have

$$\sum_{t=1}^T \lambda^{t-1} h_t(x_t^k) - \underline{f}_1^{k-1}(x_1^k) \leq \sum_{t=1}^T \lambda^{t-1} \epsilon_{t-1}. \quad (3.32)$$

Proof By the definition of $\underline{f}_1^{k-1}(x_1^k)$ in (3.9), we have

$$\underline{f}_1^{k-1}(x_1^k) = h_1(x_1^k) + \lambda \underline{v}_2^{k-1}(x_1^k),$$

which together with our assumption in (3.31) imply that

$$h_1(x_1^k) + \lambda \underline{v}_2^{k-1}(x_1^k) - \underline{f}_1^{k-1}(x_1^k) = \lambda[v_2^{k-1}(x_1^k) - \underline{v}_2^{k-1}(x_1^k)] \leq \epsilon_0. \quad (3.33)$$

Moreover, it follows from (3.9) and (3.13) that

$$\begin{aligned} h_t(x_t^k) + \lambda \underline{v}_{t+1}^{k-1}(x_t^k) &= \min \left\{ \underline{f}_t^{k-1}(x) : x \in X_t(x_{t-1}^k) \right\} \\ &\leq \min \left\{ f(x) : x \in X_t(x_{t-1}^k) \right\} = v_t(x_{t-1}^k), \end{aligned}$$

which, in view of our assumption

$$\lambda[v_{t+1}(x_t^k) - \underline{v}_{t+1}^{k-1}(x_t^k)] \leq \epsilon_{t-1},$$

then implies that

$$h_t(x_t^k) + \lambda v_{t+1}(x_t^k) \leq v_t(x_{t-1}^k) + \epsilon_{t-1} \quad (3.34)$$

for any $t = 2, \dots, T$. Multiplying λ^{t-1} to both side of the above inequalities, summing them up with the inequalities in (3.33), and using the fact that $v_{T+1}(x_T^k) = 0$, we have

$$\sum_{t=1}^T \lambda^{t-1} h_t(x_t^k) - \underline{f}_1^{k-1}(x_1^k) \leq \sum_{t=1}^T \lambda^{t-1} \epsilon_{t-1}.$$

□

In the sequel, we use S_t^{k-1} to denote the set of ϵ_t -saturated search points at stage t that have been generated by the algorithm before the k -th iteration. Using these sets, we now define the notion of *distinguishable* search points as follows.

Definition 2 We say that a search point x_t^k at stage t is δ_t -distinguishable if

$$g_t^k(x_t^k) > \delta_t, \quad (3.35)$$

where $g_t^k(x)$ denotes the distance between x to the set S_t^{k-1} given by

$$g_t^k(x) = \begin{cases} \min_{s \in S_t^{k-1}} \|s - x\|, & t < T, \\ 0, & \text{o.w.} \end{cases}$$

Below we show that each iteration of the DDP method will either find an ϵ_0 -solution of problem (3.1), or find a new ϵ_t -saturated and δ_t -distinguishable search point at some stage t by properly specifying δ_t and ϵ_t for $t = 0, \dots, T-1$.

Proposition 3 *Assume that $\delta_t \in [0, +\infty)$ for $t = 1, \dots, T$ are given. Also let us denote*

$$\epsilon_t := \begin{cases} 0, & t = T-1, \\ \sum_{\tau=t}^{T-2} [(M_{\tau+1} + \underline{M}_{\tau+1})\delta_{\tau+1}\lambda^{\tau-t}], & t \leq T-2. \end{cases} \quad (3.36)$$

Then, every iteration k of the DDP method will either generate a δ_t -distinguishable and ϵ_t -saturated search point x_t^k at some stage $t = 1, \dots, T$, or find a feasible policy (x_1^k, \dots, x_T^k) of problem (3.1) such that

$$f_1(x_1^k) - f^* \leq \epsilon_0, \quad (3.37)$$

$$\sum_{t=1}^T \lambda^{t-1} h_t(x_t^k) - \underline{f}_1^{k-1}(x_1^k) \leq \sum_{t=1}^T \lambda^{t-1} \epsilon_{t-1}. \quad (3.38)$$

Proof First note that the definition of ϵ_t is computed according to the recursion $\epsilon_{t-1} = (M_t + \underline{M}_t)\delta_t + \lambda\epsilon_t$ (see (3.24)) and the assumption that $\epsilon_{T-1} = 0$. Next, observe that exactly one of the following T cases will happen at the k -th iteration of the DDP method.

Case 1: $g_t^k(x_t^k) \leq \delta_t, \forall 1 \leq t \leq T-1$;

Case t , $t = 2, \dots, T-1$: $g_t^k(x_t^k) \leq \delta_t, \forall t \leq i \leq T-1$, and $g_{t-1}^k(x_{t-1}^k) > \delta_{t-1}$;

Case T : $g_{T-1}^k(x_{T-1}^k) > \delta_{T-1}$.

We start with the first case. In this case, we have $g_t^k(x_t^k) \leq \delta_t, \forall 1 \leq t \leq T-1$. Hence, x_t^k must be close to an existing ϵ_t -saturated point $x_t^{j_t}$ for some $j_t \leq k-1$ s.t.

$$\|x_t^k - x_t^{j_t}\| \leq \delta_t, \quad \forall 1 \leq t \leq T-1. \quad (3.39)$$

It then follows from the above relation (with $t = 1$), (3.24), and the fact $f^* \geq \underline{f}_1^{k-1}(x_1^k)$ that

$$f_1(x_1^k) - f^* \leq f_1(x_1^k) - \underline{f}_1^{k-1}(x_1^k) = \lambda[v_2(x_1^k) - \underline{v}_2^{k-1}(x_1^k)] \leq \epsilon_0. \quad (3.40)$$

Moreover, we conclude from (3.24) and (3.39) that

$$\lambda[v_{t+1}(x_t^k) - \underline{v}_{t+1}^{k-1}(x_t^k)] \leq \epsilon_{t-1}, \quad \forall 1 \leq t \leq T-1. \quad (3.41)$$

Hence, the assumptions in Lemma 4 hold and the result in (3.38) immediately follows.

We now examine the t -th case for any $2 \leq t \leq T-2$. In these cases, we have $g_{t-1}^k(x_{t-1}^k) > \delta_{t-1}$ and thus x_{t-1}^k is δ_{t-1} -distinguishable. In addition, we have $g_t^k(x_t^k) \leq \delta_t$. As a result, x_t^k must be close to an existing ϵ_t -saturated point $x_t^{j_t}$ with $j_t \leq k-1$. This observation, in view of (3.25), then implies that $v_t(x_{t-1}^k) - \underline{v}_t^k(x_{t-1}^k) \leq \epsilon_{t-1}$. Hence x_{t-1}^k is both δ_{t-1} -distinguishable and ϵ_{t-1} -saturated.

For the T -th case, we have $g_{T-1}^k(x_{T-1}^k) > \delta_{T-1}$ and hence x_{T-1}^k is δ_t -distinguishable. Also by Lemma 3, x_{T-1}^k will get 0-saturated. Therefore, x_{T-1}^k is δ_{T-1} -distinguishable and ϵ_{T-1} -saturated (with $\epsilon_{T-1} = 0$).

Combining all these cases together, we conclude that every DDP iteration will either generate a δ_t -distinguishable and ϵ_t -saturated search point at some stage $t = 1, \dots, T$, or find a feasible policy of problem (3.1) satisfying (3.37) and (3.38). \square

It is worth noting that each DDP iteration can possibly generate more than one δ_t -distinguishable and ϵ_t -saturated points. For example, for the t -case in the the above proof of Proposition 3, we pointed out that x_{t-1}^k is δ_{t-1} -distinguishable and ϵ_{t-1} -saturated. Some other search point x_i^k with $i \leq t-2$ in the preceding stages might also become δ_i -distinguishable and ϵ_i -saturated even though there are no such guarantees.

We are now ready to establish the complexity of the DDP method. For the sake of simplicity, we will fix the norm $\|\cdot\|$ to be an l_∞ norm to define the distances and Lipschitz constants at each stage t . It should be noted, however, that the DDP method itself does not really depend on the selection of norms. The l_∞ norm is chosen because it will help us to count the number of search points needed in each stage to guarantee the convergence of the algorithm.

Theorem 1 Suppose that the norm used to define the bound on D_t in (3.6) is the l_∞ norm. Also assume that $\delta_t \in [0, +\infty)$ are given and that ϵ_t are defined in (3.36). Then the number of iterations performed by the DDP method to find a solution satisfying (3.37) and (3.38) can be bounded by

$$\sum_{t=1}^{T-1} \left(\frac{D_t}{\delta_t} + 1 \right)^{n_t} + 1. \quad (3.42)$$

In particular, If $n_t \leq n$, $D_t \leq D$, $\max\{M_t, \underline{M}_t\} \leq M$ and $\delta_t = \epsilon$ for all $t = 1, \dots, T$, then the DDP method will find a feasible policy (x_1^k, \dots, x_T^k) of problem (3.1) s.t.

$$f_1(x_1^k) - f^* \leq 2M \min \left\{ \frac{1}{1-\lambda}, T-1 \right\} \epsilon, \quad (3.43)$$

$$\sum_{t=1}^T \lambda^{t-1} h_t(x_t^k) - \underline{f}_1^{k-1}(x_1^k) \leq 2M \min \left\{ \frac{1}{(1-\lambda)^2}, \frac{T(T-1)}{2} \right\} \epsilon \quad (3.44)$$

within at most

$$(T-1) \left(\frac{D}{\epsilon} + 1 \right)^n + 1 \quad (3.45)$$

iterations.

Proof Let us count the total number of possible search points for saturation before a solution satisfying (3.37) and (3.38) is found. Using (3.35) and the assumption the effective feasible region for each stage t is inside a box with side length D_t (c.f., (3.6)), we can see that the number of possible δ_t -distinguishable search points for saturation at each stage is given by

$$N_t := \left(\frac{D_t}{\delta_t} + 1 \right)^{n_t}.$$

This observation together with Proposition 3 then imply that the total number of iterations performed by DDP will be bounded by $\sum_{t=1}^{T-1} N_t + 1$ and hence by (3.42).

Now suppose that $n_t \leq n$, $D_t \leq D$, $\max\{M_t, \underline{M}_t\} \leq M$ and $\delta_t = \epsilon$ for all $t = 1, \dots, T$. We first provide a bound on ϵ_t defined in (3.36). For $0 \leq t \leq T-2$, we have

$$\begin{aligned} \epsilon_t &= \sum_{\tau=t}^{T-2} \lambda^{\tau-t} [(M_{\tau+1} + \underline{M}_{\tau+1}) \delta_{\tau+1}] \\ &= 2M \sum_{\tau=t}^{T-2} \lambda^{\tau-t} \epsilon \\ &\leq 2M \min \left\{ \frac{1-\lambda^{T-t-1}}{1-\lambda}, T-t-1 \right\} \epsilon \\ &\leq 2M \min \left\{ \frac{1}{1-\lambda}, T-t-1 \right\} \epsilon, \end{aligned} \quad (3.46)$$

and as a result,

$$\begin{aligned} \sum_{t=1}^T \lambda^{t-1} \epsilon_{t-1} &= \sum_{t=1}^T \lambda^{t-1} \epsilon_{t-1} = \sum_{t=0}^{T-2} \lambda^t \epsilon_t \\ &\leq 2M \sum_{t=0}^{T-2} \left[\lambda^t \min \left\{ \frac{1}{1-\lambda}, T-t-1 \right\} \right] \epsilon \\ &\leq 2M \sum_{t=0}^{T-2} \min \left\{ \frac{\lambda^t}{1-\lambda}, T-t-1 \right\} \epsilon \\ &\leq 2M \min \left\{ \frac{1}{(1-\lambda)^2}, \frac{T(T-1)}{2} \right\} \epsilon. \end{aligned} \quad (3.47)$$

Using these bounds in (3.37) and (3.38), we obtain relations (3.43) and (3.44). Moreover, the iteration complexity bound in (3.45) follows directly from (3.42). \square

We now add some remarks about the results obtained in Theorem 1.

Firstly, similar to the basic cutting plane method, the bound in (3.42) has an exponential dependence on n_t . However, since the algorithm itself does not require us to explicitly discretize the decision variables in \mathbb{R}^{n_t} , the complexity bound actually depends on the dimension of the affine space spanned by effective feasible region \mathcal{X}_t defined in (3.5), which can be smaller than the nominal dimension n_t .

Secondly, it is interesting to examine the dependence of the complexity bound in (3.45) on the number of stages T . In particular, if the discounting factor $\lambda < 1$,

the number of iterations required to find an ϵ -solution of problem (3.1), i.e., a point \bar{x}_1 s.t. $f_1(\bar{x}_1) - f^* \leq \epsilon$ only linearly depends on T . When the discounting factor $\lambda = 1$, we can see that T also appears in the termination criterions (3.43) and (3.44). As a result, the number of iterations required to find an ϵ -solution of problem (3.1) will depend on T^n . The discounting factor provides a mechanism to compensate the errors accumulated from approximating the value function v_{t+1} by v_{t+1}^k starting from $t = T - 1$ to $t = 1$.

Thirdly, while the termination criterion in (3.43) cannot be verified since the function value f_1 and f^* are not easily computable, the gap between the upper and lower bound in the l.h.s. of (3.44) can be computed as we run the algorithm. It should be noted that the dependence on T for these two criterions are slightly different especially when the discounting factor $\lambda = 1$ (see the r.h.s. of (3.43) and (3.44)).

4 Explorative dual dynamic programming

In this section, we generalize the DDP method for solving the multi-stage stochastic optimization problems which have potentially an exponential number of scenarios. As discussed in Sect. 1, we assume that we can sample from the probability distribution P_t of the random vector ξ_t , $t = 2, \dots, T$. A sample average approximation (SAA) of the original problem (1.1) is constructed by replacing the true distribution of $\xi_t = (A_t, b_t, B_t, Q_t, p_t, c_t)$ with the empirical distribution P_{N_t} based on a random sample

$$\tilde{\xi}_t = (\tilde{A}_{ti}, \tilde{b}_{ti}, \tilde{B}_{ti}, \tilde{Q}_{ti}, \tilde{p}_{ti}, \tilde{c}_{ti}), i = 1, \dots, N_t$$

from the distribution P_t of size N_t . Consequently the probability distribution $P_2 \times \dots \times P_T$ of the random process ξ_2, \dots, ξ_T is replaced by $P_{N_2} \times \dots \times P_{N_T}$. Under the stage-wise independence assumption of P_t and hence P_{N_t} , it has been shown in [27] that under mild regularity assumptions we can approximate problem (1.4) by the SAA problem defined as

$$F^* := \min_{x_1 \in X_1} \{F_{11}(x_1) := H_1(x_1, c_1) + \lambda V_2(x_1)\}, \quad (4.1)$$

where the value functions V_t , $t = 2, \dots, T$, are recursively defined by

$$\begin{aligned} V_t(x_{t-1}) &:= \frac{1}{N_t} \sum_{i=1}^{N_t} v_{ti}(x_{t-1}), \\ v_{ti}(x_{t-1}) &:= \min_{x_t \in X_t(x_{t-1}, \tilde{\xi}_{ti})} \{F_{ti}(x_t) := H_t(x_t, \tilde{c}_{ti}) + \lambda V_{t+1}(x_t)\}, \end{aligned} \quad (4.2)$$

and

$$V_{T+1}(x_T) = 0. \quad (4.3)$$

We will focus on how to solve the SAA problem in (4.1). The essential difference between this problem and the single-scenario problem in (3.1) is that each stage t involves N_t (rather than one) subproblems. As a consequence, when determining the search point x_t^k at each stage t in the forward phase, we need to choose one out of N_t

feasible solutions and each one of them corresponds to a realization $\tilde{\xi}_{ti}$ of the random variables. In this section, we will present a deterministic dual dynamic programming method which chooses the feasible solution in the forward phase in an aggressive manner, while in next section, we will discuss a stochastic approach in which the feasible solution in the forward phase will be chosen randomly. As we will see, the former approach will exhibit better iteration complexity while the latter one is easier to implement. We start with the deterministic approach also because the analysis for the latter stochastic method is built on the one for the deterministic approach.

Let \mathcal{X}_{ti} be the effective feasible region for the i -th subproblem in stage t , and $\bar{\mathcal{X}}_t$ be the effective feasible region all the subproblems in stage t , respectively, given by

$$\mathcal{X}_{ti} := \begin{cases} X_1, & t = 1, \\ \cup_{x \in \bar{\mathcal{X}}_{t-1}} X_t(x, \tilde{\xi}_{ti}), & t \geq 2, \end{cases}$$

and

$$\bar{\mathcal{X}}_t := \begin{cases} X_1, & t = 1, \\ \cup_{i=1, \dots, N_t} \mathcal{X}_{ti}, & t \geq 2. \end{cases}$$

Observe that $\bar{\mathcal{X}}_t$ is not necessarily convex. Moreover, letting $\text{Aff}(\bar{\mathcal{X}}_t)$ be the affine hull of $\bar{\mathcal{X}}_t$ and $\mathcal{B}_t(\epsilon) := \{y \in \text{Aff}(\bar{\mathcal{X}}_t) : \|y\| \leq \epsilon\}$, we use

$$\bar{\mathcal{X}}_t(\epsilon) := \bar{\mathcal{X}}_t + \mathcal{B}_t(\epsilon)$$

to denote $\bar{\mathcal{X}}_t$ together with its small surrounding neighborhood.

We make the following assumptions throughout this section.

Assumption 4 For any $t \geq 1$, there exists $D_t \geq 0$ s.t.

$$\|x_t - x'_t\| \leq D_t, \quad \forall x_t, x'_t \in \bar{\mathcal{X}}_t, \quad \forall t \geq 1. \quad (4.4)$$

With a little abuse of notation, we still use D_t as in the previous section to bound the “diameter” of the effective feasible region $\bar{\mathcal{X}}_t$. Clearly, Assumption 4 holds if the convex sets $\bar{\mathcal{X}}_t$ are compact, since by definition we have $\bar{\mathcal{X}}_t \subseteq \text{Conv}(\bar{\mathcal{X}}_t) \subseteq \bar{\mathcal{X}}_t$, $\forall t \geq 1$.

Assumption 5 For any $t \geq 1$, there exists $\bar{\epsilon}_t \in (0, +\infty)$ s.t.

$$H_t(x, \tilde{\xi}_{ti}) < +\infty, \quad \forall x \in \bar{\mathcal{X}}_t(\bar{\epsilon}_t), \quad \forall i = 1, \dots, N_t, \quad (4.5)$$

$$\text{rint}\left(X_{t+1}(x, \tilde{\xi}_{(t+1)i})\right) \neq \emptyset, \quad \forall x \in \bar{\mathcal{X}}_t(\bar{\epsilon}_t), \quad \forall i = 1, \dots, N_{t+1}, \quad (4.6)$$

where $\text{rint}(\cdot)$ denotes the relative interior of a convex set.

Assumption 5 describes certain regularity conditions of problem (4.1). Specifically, the conditions in (4.5) and (4.6) imply that $H_t(x, \tilde{\xi}_{ti})$ and V_{t+1} are finitely valued in

a small neighborhood of \mathcal{X}_{ti} . The second relation in (3.7) also implies the Slater condition of the feasible sets in (4.2) and thus the existence of optimal dual solutions to define the cutting plane models for problem (4.1). Here the relative interior is required due to the nonlinearity of the constraint functions in (4.2) and we can replace $\text{rint}\left(X_{t+1}(x, \tilde{\xi}_{(t+1)i})\right)$ with $X_{t+1}(x, \tilde{\xi}_{(t+1)i})$ if the latter is polyhedral.

In view of Assumption 5, the objective functions F_{ti} must be Lipschitz continuous over $\mathcal{X}_{ti}(\bar{\epsilon}_t)$. We explicitly state the Lipschitz constants of F_{ti} below since they will be used in the convergence analysis of our algorithms. For the sake of notation convenience, we still use M_t to denote the Lipschitz constants for F_{ti} .

Assumption 6 For any $t \geq 1$ and $i = 1, \dots, N_t$, there exists $M_t \geq 0$ s.t.

$$|F_{ti}(x_t) - F_{ti}(x'_t)| \leq M_t \|x_t - x'_t\|, \quad \forall x_t, x'_t \in \mathcal{X}_{ti}. \quad (4.7)$$

We now formally state the explorative dual dynamic programming (EDDP) method as shown in Algorithm 3. A distinctive feature of EDDP is that it maintains a set of saturated search points S_t^k for each stage t . Similar to Definition 1, we say that a search point x_t^k generated by the EDDP method is ϵ_t -saturated at iteration k if

$$V_{t+1}(x_t^k) - \underline{V}_{t+1}^k(x_t^k) \leq \epsilon_t. \quad (4.14)$$

Moreover, similar to Definition 2, we say an ϵ_t -saturated search point x_t^k at stage t is δ_t -distinguishable if

$$\|x_t^k - x_t^j\| > \delta_t$$

for all other ϵ_t -saturated search points x_t^j that have been generated for stage t so far by the algorithm. Equivalently, an ϵ_t -saturated search point x_t^k is δ_t -distinguishable if

$$g_t^k(x_t^k) > \delta_t. \quad (4.15)$$

Here $g_t^k(x_t^k)$ (c.f., (4.9)) denotes the distance between x_t^k to the set S_t^{k-1} , i.e., the set of currently saturated search points in stage t . Similar to the DDP method, saturation is defined for two given related sequences $\{\epsilon_t\}$ and $\{\delta_t\}$. More precisely, the proposed algorithm takes $\{\delta_t\}$ as an initial argument and ends with $\{\epsilon_t\}$ (derived from $\{\delta_t\}$) saturated points.

In the forward phase of EDDP, for each stage t , we solve N_t subproblems as shown in (4.8) to compute the search points \tilde{x}_{ti}^k , $i = 1, \dots, N_t$. For each \tilde{x}_{ti}^k , we further compute the quantity $g_t^k(\tilde{x}_{ti}^k)$ in (4.9), i.e., the distance between \tilde{x}_{ti}^k and the set S_t^{k-1} of currently saturated search points in stage t . Then we will choose from \tilde{x}_{ti}^k , $i = 1, \dots, N_t$, the one with the largest value of $g_t^k(\tilde{x}_{ti}^k)$ as x_t^k , i.e., $g_t^k(x_t^k) = \max_{i=1, \dots, N_t} g_t^k(\tilde{x}_{ti}^k)$. We can break the ties arbitrarily (or randomly to be consistent with the algorithm in the next section). The search point x_t^k is deemed to be saturated if $g_t^k(x_t^k)$ is small enough, therefore so is the case for \tilde{x}_{ti}^k for all i . As a consequence, the point x_{t-1}^k must also be saturated and can be added to S_{t-1}^k . We call the sequence (x_1^k, \dots, x_T^k) a forward path

Algorithm 3 Explorative dual dynamic programming (EDDP)

1: Set $V_t^0(x) = -\infty$, $t = 2, \dots, T$, $\underline{V}_{T+1}^k(x) = 0$, $k \geq 1$, and $S_t^0 = \emptyset$, $t = 1, \dots, T$.
2: **for** $k = 1, 2, \dots$, **do**
3: **for** $t = 1, \dots, T$ **do** ▷ Forward phase.
4: **for** $i = 1, 2, \dots, N_t$ **do**

$$\tilde{x}_{ti}^k \in \operatorname{Argmin}_{x \in X_t(x_{t-1}^k, \tilde{\xi}_{ti})} \left\{ \underline{F}_{ti}^{k-1}(x) := H_t(x, \tilde{c}_{ti}) + \lambda \underline{V}_{t+1}^{k-1}(x) \right\}. \quad (4.8)$$

$$g_t^k(\tilde{x}_{ti}^k) = \begin{cases} \min_{s \in S_t^{k-1}} \|s - \tilde{x}_{ti}^k\|, & t < T, \\ 0, & \text{o.w.} \end{cases} \quad (4.9)$$

5: **end for**
6: Choose x_t^k from $\{\tilde{x}_{ti}^k\}$ such that $g_t^k(x_t^k) = \max_{i=1, \dots, N_t} g_t^k(\tilde{x}_{ti}^k)$.
7: **end for**
8: **if** $g_1^k(x_1^k) \leq \delta_0$ **then** Terminate.
9: **for** $t = T, T-1, \dots, 2$ **do** ▷ Backward phase.
10: **if** $g_t^k(x_t^k) \leq \delta_t$ **then**
11: Set $S_{t-1}^k = S_{t-1}^{k-1} \cup \{x_{t-1}^k\}$.
12: **end if**
13: **for** $i = 1, \dots, N_t$ **do**

$$\tilde{v}_{ti}^k(x_{t-1}^k) = \min_{x \in X_t(x_{t-1}^k, \tilde{\xi}_{ti})} \left\{ \underline{F}_{ti}^k(x) := H_t(x, \tilde{c}_{ti}) + \lambda \underline{V}_{t+1}^k(x) \right\}. \quad (4.10)$$

$$(\tilde{v}_{ti}^k)'(x_{t-1}^k) = [\tilde{B}_{ti}, \tilde{Q}_{ti}] y_{ti}^k, \text{ where } y_{ti}^k \text{ is the optimal dual multipliers of (4.10).} \quad (4.11)$$

14: **end for**
15: **end for**

$$\tilde{V}_t^k = \frac{1}{N_t} \sum_{j=1}^{N_t} \tilde{v}_{tj}^k(x_{t-1}^k), (\tilde{V}_t^k)' = \frac{1}{N_t} \sum_{j=1}^{N_t} (\tilde{v}_{tj}^k)'(x_{t-1}^k). \quad (4.12)$$

$$V_t^k(x) = \max \left\{ \underline{V}_t^{k-1}(x), \tilde{V}_t^k + \langle (\tilde{V}_t^k)', x - x_{t-1}^k \rangle \right\}. \quad (4.13)$$

16: **end for**
17: **end for**

at iteration k , since it is the trajectory generated in the forward phase for one particular scenario of the data process $\tilde{\xi}_{ti}$. In view of the above discussion, the EDDP method always chooses the most “distinguishable” forward path to encourage exploration in an aggressive manner (see Line 6 of Algorithm 3). This also explains the origin of the name EDDP.

The backward phase of EDDP is similar to the DDP in Algorithm 2 with the following differences. First, we need to update the set S_t^k for the saturated search points. Second, the computation of the cutting plane model also requires the solutions of N_t subproblems in (4.10).

The following result is similar to Lemma 1 for the DDP method.

Lemma 5 For any $k \geq 1$,

$$\underline{V}_t^{k-1}(x) \leq \underline{V}_t^k(x) \leq \frac{1}{N_t} \sum_{j=1}^{N_t} \tilde{v}_{tj}^k(x) \leq V_t(x), \forall x \in \bar{\mathcal{X}}_{t-1}(\bar{\epsilon}_{t-1}), t = 2, \dots, T, \quad (4.16)$$

$$\underline{F}_{ti}^{k-1}(x) \leq \underline{F}_{ti}^k(x) \leq F_{ti}(x), \forall x \in \bar{\mathcal{X}}_t(\bar{\epsilon}_t), t = 1, \dots, T, i = 1, \dots, N_t. \quad (4.17)$$

Proof The proof is similar to that of Lemma 1. The major difference exists in that (3.15) will be replaced by

$$\begin{aligned} \underline{V}_{t-1}^k(x) &= \frac{1}{N_{t-1}} \sum_{j=1}^{N_{t-1}} \left[\tilde{v}_{(t-1)j}^1(x_{t-2}^k) + \langle (\tilde{v}_{(t-1)j}^k)'(x_{t-2}^k), x - x_{t-2}^k \rangle \right] \\ &\leq \frac{1}{N_{t-1}} \sum_{j=1}^{N_{t-1}} \tilde{v}_{(t-1)j}^k(x) \leq \frac{1}{N_{t-1}} \sum_{j=1}^{N_{t-1}} v_{(t-1)j}^k(x) \\ &= V_{t-1}(x), \end{aligned}$$

and hence we skip the details. \square

In order to establish the complexity of the EDDP Algorithm, we need to show that the approximation functions $\underline{F}_{ti}^k(\cdot)$ are Lipschitz continuous on \mathcal{X}_{ti} . For convenience, we still use \underline{M}_t to denote the Lipschitz constants for \underline{F}_{ti}^k . We skip its proof since it is similar to that of Lemma 2 after replacing Assumption 2 with Assumption 5.

Lemma 6 For any $t \geq 1$ and $i = 1, \dots, N_t$, there exists $\underline{M}_t \geq 0$ s.t.

$$|\underline{F}_{ti}^k(x_t) - \underline{F}_{ti}^k(x'_t)| \leq \underline{M}_t \|x_t - x'_t\|, \quad \forall x_t, x'_t \in \bar{\mathcal{X}}_t(\bar{\epsilon}_t) \quad \forall k \geq 1. \quad (4.18)$$

Below we describe some basic properties about the saturation of search points.

Lemma 7 Any search point x_{T-1}^k generated for the $(T-1)$ -th stage in EDDP must be 0-saturated for any $k \geq 1$.

Proof Note that by (4.16), we have $\underline{V}_T^k(x_{T-1}^k) \leq V(x_{T-1}^k)$. Moreover, by (4.13),

$$\begin{aligned} \underline{V}_T^k(x_{T-1}^k) &\geq \frac{1}{N_T} \sum_{j=1}^{N_T} \left[\tilde{v}_{tj}^k(x_{T-1}^k) + \langle (\tilde{v}_{tj}^k)'(x_{T-1}^k), \tilde{\xi}_{tj} \rangle, x_{T-1}^k - x_{T-1}^k \rangle \right] \\ &= \frac{1}{N_T} \sum_{j=1}^{N_T} \tilde{v}_{tj}^k(x_{T-1}^k) = \frac{1}{N_T} \sum_{j=1}^{N_T} v_{tj}(x_{T-1}^k) \\ &= V(x_{T-1}^k) \end{aligned}$$

where the second-to-last equality follows from the fact that $v_{T+1}^k = 0$ and the definitions of $v_{Tj}(x)$ and $\tilde{v}_{Tj}^k(x)$ in (4.2) and (4.10). Therefore we must have $\underline{V}_T^k(x_{T-1}^k) = V(x_{T-1}^k)$, which, in view of (4.14), implies that x_T^k is 0-saturated. \square

We now generalize the result in Proposition 2 for the DDP method to relate the saturation of search points across two consecutive stages in the EDDP method.

Proposition 4 *Assume that $\delta_t \in [0, +\infty)$ for $t = 1, \dots, T$ are given and that ϵ_t are defined recursively according to (3.24) for some given $\epsilon_{T-1} > 0$. Also let $g_t^k(\cdot)$ be defined in (4.9) and assume that x_t^k is chosen such that*

$$g_t^k(x_t^k) = \max_{i=1, \dots, N_t} g_t^k(\tilde{x}_{ti}^k).$$

(a) *If $g_t^k(x_t^k) \leq \delta_t$, $t = 2, \dots, T-1$, then we have*

$$F_{ti}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{k-1}(\tilde{x}_{ti}^k) = \lambda[V_{t+1}(\tilde{x}_{ti}^k) - \underline{V}_{t+1}^{k-1}(\tilde{x}_{ti}^k)] \leq \epsilon_{t-1}. \quad (4.19)$$

Moreover, for any $T \geq 2$, we have

$$V_t(x_{t-1}^k) - \underline{V}_t(x_{t-1}^k) \leq \epsilon_{t-1}. \quad (4.20)$$

where ϵ_{t-1} is defined (3.24).

(b) S_t^k , $t = 1, \dots, T-1$, contains all the ϵ_t -saturated search points at stage t generated by the algorithm up to the k -th iteration.

Proof We prove the results by induction. First note that by (4.9) we have $g_T^k(x_T^k) = 0$. Moreover, by Lemma 7, any search point x_{T-1}^k will be 0-saturated and hence part a) holds with $\epsilon_{T-1} = 0$ for $t = T-1$. Moreover, in view of Line 11 of Algorithm 3 and the fact $g_T^k(x_T^k) = 0$, S_{T-1}^k contains all the 0-saturated search point obtained for stage $T-1$ and hence part b) holds for $t = T-1$.

Now assume that $g_t^k(x_t^k) \leq \delta_t$ for the t -th stage for some $t \leq T-1$. In view of this assumption and the definition of x_t^k , we have

$$g_t^k(\tilde{x}_{ti}^k) = \min_{s \in S_t^{k-1}} \|s - \tilde{x}_{ti}^k\| \leq \delta_t$$

for any $i = 1, \dots, N_t$. Note that we must have $S_t^{k-1} \neq \emptyset$ since otherwise $g_t^k(\tilde{x}_{ti}^k) = +\infty$. Hence, there exists $x_t^{j_i} \in S_t^{k-1}$ for some $j_i < k-1$ such that

$$\|x_t^{j_i} - \tilde{x}_{ti}^k\| \leq \delta_t, \quad (4.21)$$

$$V_{t+1}(x_t^{j_i}) - \underline{V}_{t+1}^{j_i}(x_t^{j_i}) \leq \epsilon_t, \quad (4.22)$$

for any $t = 1, \dots, N_t$.

Observe that by the definition of x_{ti}^k in (4.8) and the first relation in (4.17), we have

$$\begin{aligned} F_{ti}(\tilde{x}_{ti}^k) - \min_{x \in X_{ti}(x_{t-1}^k)} \underline{F}_{ti}^{k-1}(x) &= F_{ti}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{k-1}(\tilde{x}_{ti}^k) \\ &\leq F_{ti}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{j_i}(\tilde{x}_{ti}^k). \end{aligned} \quad (4.23)$$

Moreover, by (4.7) and (4.18), we have

$$|F_{ti}(\tilde{x}_{ti}^k) - F_{ti}(x_t^{j_i})| \leq M_t \|\tilde{x}_{ti}^k - x_t^{j_i}\| \text{ and } |\underline{F}_{ti}^{j_i}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{j_i}(x_t^{j_i})| \leq \underline{M}_t \|\tilde{x}_{ti}^k - x_t^{j_i}\|.$$

In addition, it follows from the definitions of F_{ti} and \underline{F}_{ti}^k (c.f. (4.2) and (4.8)) and (4.22) that

$$F_{ti}(x_t^{j_i}) - \underline{F}_{ti}^{j_i}(x_t^{j_i}) = \lambda[V_{t+1}(x_t^{j_i}) - \underline{V}_{t+1}^{j_i}(x_t^{j_i})] \leq \lambda\epsilon_t.$$

Combining the previous observations and (4.22), we have

$$\begin{aligned} & F_{ti}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{k-1}(\tilde{x}_{ti}^k) \\ & \leq [F_{ti}(\tilde{x}_{ti}^k) - F_{ti}(x_t^{j_i})] + [F_{ti}(x_t^{j_i}) - \underline{F}_{ti}^{j_i}(x_t^{j_i})] + [\underline{F}_{ti}^{j_i}(x_t^{j_i}) - \underline{F}_{ti}^{j_i}(\tilde{x}_{ti}^k)] \\ & \leq (M_t + \underline{M}_t) \|\tilde{x}_{ti}^k - x_t^{j_i}\| + \lambda\epsilon_t \\ & \leq (M_t + \underline{M}_t) \delta_t + \lambda\epsilon_t = \epsilon_{t-1}, \end{aligned} \quad (4.24)$$

where the last inequality follows from the definition of ϵ_{t-1} in (3.24). The above result, in view of the definitions of F_{ti} and \underline{F}_{ti}^k , then implies (4.19).

We will now show that the search point x_{t-1}^k in the preceding stage $t-1$ must also be ϵ_{t-1} -saturated at iteration k . Note that \tilde{x}_{ti}^k are feasible solutions for the t -th stage problem and hence that the function value $F_{ti}(\tilde{x}_{ti}^k)$ must be greater than the optimal value $v_{ti}(x_{t-1}^k)$ defined in (4.2). Using this observation, we have

$$\begin{aligned} V_t(x_{t-1}^k) - \underline{V}_t^k(x_{t-1}^k) &= \frac{1}{N_t} \sum_{i=1}^{N_t} v_{ti}(x_{t-1}^k) - \underline{V}_t^k(x_{t-1}^k) \\ &\leq \frac{1}{N_t} \sum_{i=1}^{N_t} F_{ti}(\tilde{x}_{ti}^k) - \underline{V}_t^k(x_{t-1}^k). \end{aligned} \quad (4.25)$$

Moreover, using the definitions of $\underline{V}_t^k(x_{t-1}^k)$ and $\tilde{v}_{ti}^k(x_{t-1}^k)$ in (4.13) and (4.10), the relations in (4.16) and the fact that $\underline{V}_{t+1}^k(x) \geq \underline{V}_{t+1}^{k-1}(x)$ due to (4.17), we have

$$\begin{aligned} \underline{V}_t^k(x_{t-1}^k) &= \max\{\underline{V}_t^{k-1}(x_{t-1}^k), \frac{1}{N_t} \sum_{i=1}^{N_t} \tilde{v}_{ti}^k(x_{t-1}^k)\} \\ &= \frac{1}{N_t} \sum_{i=1}^{N_t} \tilde{v}_{ti}^k(x_{t-1}^k) \\ &= \frac{1}{N_t} \sum_{i=1}^{N_t} \min \left\{ \underline{F}_{ti}^k(x) : x \in X_t(x_{t-1}^k, \tilde{\xi}_{tj}) \right\} \\ &\geq \frac{1}{N_t} \sum_{i=1}^{N_t} \min \left\{ \underline{F}_{ti}^{k-1}(x) : x \in X_t(x_{t-1}^k, \tilde{\xi}_{tj}) \right\} \end{aligned}$$

$$= \frac{1}{N_t} \sum_{i=1}^{N_t} \underline{F}_{ti}^{k-1}(\tilde{x}_{ti}^k), \quad (4.26)$$

where the last identity follows from the definition of x_i^k in (4.8). Putting together (4.25) and (4.26), we have

$$\begin{aligned} V_t(x_{t-1}^k) - \underline{V}_t^k(x_{t-1}^k) &\leq \frac{1}{N_t} \sum_{i=1}^{N_t} [F_{ti}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{k-1}(\tilde{x}_{ti}^k)] \\ &\leq \epsilon_{t-1}, \end{aligned} \quad (4.27)$$

where the last inequality follows from (4.24). The above inequality then implies that x_{t-1}^k gets saturated at the k -th iteration. Moreover, the point x_{t-1}^k will be added into the set S_{t-1}^k in view of the definition in Line 11 of Algorithm 3. We have thus shown both part (a) and part (b). \square

Different from the DDP method, we do not have a convenient way to compute an exact upper bound on the optimal value for the general multi-stage stochastic optimization problem. However, we can use $g_1^k(x_1^k)$ as a termination criterion for the EDDP method. Indeed, using (4.24) (with $t = 1$ and $i = 1$) and the fact that $N_1 = 1$, we conclude that if $g_1^k(x_1^k) \leq \delta_1$, then we must have

$$F_{11}(x_1^k) - F^* \leq F_{11}(x_1^k) - \underline{F}_{11}^{k-1}(x_1^k) \leq \epsilon_0. \quad (4.28)$$

It is worth noting that one can possibly provide a stochastic upper bound on F^* for solving multi-stage stochastic optimization problems. We will discuss this idea further in Sect. 5.

Below we show that each iteration of the EDDP method will either find an ϵ_0 -solution of problem (4.1), or find a new ϵ_t -saturated and δ_t -distinguishable search point at some stage t .

Proposition 5 *Assume that $\delta_t \in [0, +\infty)$, $t = 1, \dots, T$, are given. Also let ϵ_t , $t = 0, \dots, T$, be defined in (3.36). Then any iteration k of the EDDP method will either generate a new ϵ_t -saturated and δ_t -distinguishable search point x_t^k at some stage $t = 1, \dots, T$, or find a feasible solution x_1^k of problem (4.1) such that*

$$F_{11}(x_1^k) - F^* \leq \epsilon_0. \quad (4.29)$$

Proof Similar to the proof of Proposition 3, we consider the following T cases that will happen at the k -th iteration of the EDDP method.

Case 1: $g_t^k(x_t^k) \leq \delta_t$, $\forall 1 \leq t \leq T-1$;

Case t , $t = 2, \dots, T-1$: $g_t^k(x_t^k) \leq \delta_t$, $\forall t \leq i \leq T-1$, and $g_{t-1}^k(x_{t-1}^k) > \delta_{t-1}$;

Case T : $g_{T-1}^k(x_{T-1}^k) > \delta_{T-1}$.

For the first case, it follows from the assumption $g_1^k(x_1^k) \leq \delta_1$ and (4.28) that x_1^k must be an ϵ_0 -solution of problem (4.1). Now let us consider the t -th case for any

$t = 2, \dots, T-2$. Since $g_{t-1}^k(x_{t-1}^k) > \delta_{t-1}$, the search point x_{t-1}^k is δ_t -distinguishable. Moreover, we conclude from the assumption $g_t^k(x_t^k) \leq \delta_t$ and Proposition 4.a) that the point x_{t-1}^k must be ϵ_{t-1} -saturated. Hence, the search point x_{t-1}^k is δ_t -distinguishable and ϵ_{t-1} -saturated for the t -th case, $t = 2, \dots, T-1$. Finally for the T -th case, x_{T-1}^k is δ_{T-1} -distinguishable by assumption. Moreover, by Lemma 7, x_{T-1}^k in the $(T-1)$ -stage will get 0-saturated. Hence x_{T-1}^k is δ_{T-1} -distinguishable and ϵ_{T-1} -saturated. The result then follows by putting all these cases together. \square

We are now ready to establish the complexity of the EDDP method. For the sake of simplicity, we will fix the norm $\|\cdot\|$ to be an l_∞ norm to define the distances and Lipschitz constants at each stage t .

Theorem 2 *Suppose that the norm used to define the bound on D_t in (4.4) is the l_∞ norm. Also assume that $\delta_t \in [0, +\infty)$ are given and that ϵ_t are defined in (3.36). Then the number of iterations performed by the EDDP method to find a solution satisfying*

$$F_{11}(x_1^k) - F^* \leq \epsilon \quad (4.30)$$

can be bounded by $\bar{K} + 1$, where

$$\bar{K} := \sum_{t=1}^{T-1} \left(\frac{D_t}{\delta_t} + 1 \right)^{n_t}. \quad (4.31)$$

In particular, If $n_t \leq n$, $D_t \leq D$, $\max\{M_t, \underline{M}_t\} \leq M$ and $\delta_t = \epsilon$ for all $t = 1, \dots, T$, then the EDDP method will find a solution x_1^k of problem (4.1) s.t.

$$F_{11}(x_1^k) - F^* \leq 2M \min \left\{ \frac{1}{1-\lambda}, T-1 \right\} \epsilon, \quad (4.32)$$

within at most $\bar{K}_\epsilon + 1$ iterations with

$$\bar{K}_\epsilon := (T-1) \left(\frac{D}{\epsilon} + 1 \right)^n. \quad (4.33)$$

Proof Let us count the total number of possible search points for saturation before an ϵ -optimal policy of problem (4.1) is found. Using (4.15) and the assumption the feasible region for each stage t is inside a box with side length D_t (c.f., (4.4)), we can see that the number of possible search points for saturation at each stage is given by

$$\left(\frac{D_t}{\delta_t} + 1 \right)^{n_t}.$$

As a consequence, the total number of iterations that EDDP will perform before finding an ϵ_0 -optimal policy will be bounded by $\bar{K} + 1$. If $n_t \leq n$, $D_t \leq D$, $\max\{M_t, \underline{M}_t\} \leq M$ and $\delta_t = \epsilon$ for all $t = 1, \dots, T$, we can obtain (4.32) by using the bound (3.46) for ϵ_0 in (4.30). Moreover, the bound in (4.33) follows directly from (4.31). \square

We now add some remarks about the results obtained in Theorem 2 for the EDDP method. First, comparing with the DDP method for single-scenario problems, we can see that these two algorithms exhibit similar iteration complexity. However, the DDP method provides some guarantees on an easily computable gap between the upper and lower bound. On the other hand, we can terminate the EDDP method by using the quantity g_t^k . Second, the EDDP method requires us to maintain the set of saturated search points S_t^k and explicitly use the selected norm $\|\cdot\|$ to compute g_t^k . In the next section, we will discuss a stochastic dual dynamic programming method which can address some of these issues associated with EDDP, by sacrificing a bit on the iteration complexity bound in terms of its dependence on the number of scenarios N_t . Third, similar to the DDP method, we can replace n_t in the complexity bound of the EDDP method with the dimension of the effective region $\bar{\mathcal{X}}_t$ in (4.31).

5 Stochastic dual dynamic programming

In this section, we still consider the SAA problem (4.1) for multi-stage stochastic optimization and suppose that Assumptions 4, 5 and 6 hold throughout this section. Our goal is to establish the iteration complexity of the stochastic dual dynamic programming (SDDP) for solving this problem.

As mentioned in the previous section, when dealing with multiple scenarios in each stage t , we need to select x_t^k from \tilde{x}_{ti} , $i = 1, \dots, N_t$, defined in (4.8), where \tilde{x}_{ti} corresponds to a particular realization $\tilde{\xi}_{ti}$, $i = 1, \dots, N_t$. While the EDDP method chooses x_t^k in an aggressive manner by selecting the most “distinguishable” search points, SDDP will select x_t^k from \tilde{x}_{ti} , $i = 1, \dots, N_t$, in a randomized manner.

The SDDP method is formally described in Algorithm 4. This method still consists of the forward phase and backward phase similarly to the DDP and EDDP methods. On one hand, we can view DDP as a special case of SDDP with $N_t = 1$, $t = 1, \dots, T$. On the other hand, there exist a few essential differences between SDDP in Algorithm 4 and EDDP in Algorithm 3. First, in the forward phase of SDDP, we randomly pick up an index i_t and solve problem (5.1) to update x_t^k . Equivalently, one can view x_t^k as being randomly chosen from \tilde{x}_{ti}^k , $i = 1, \dots, N_t$, defined in (4.8) for the EDDP method. Note that we do not need to compute \tilde{x}_{ti}^k for $i \neq i_t$, even though they will be used in the analysis of the SDDP method. Hence, the computation of the forward path (x_1^k, \dots, x_T^k) in SDDP is less expensive than that in EDDP. Second, in SDDP we do not need to maintain the set of saturated search points and thus the algorithmic scheme is much simplified. However, without these sets, we will not be able to compute the quantities g_t^k as in Algorithm 3 and thus cannot perform a rigorous termination test as in EDDP. We will discuss later in this section how to provide a statistical upper bound by running the forward phase a few times.

As mentioned earlier, our goal in this section is to solve the SAA problem in (4.1) instead of the original problem in (1.1). Hence the randomness for the SDDP method in Algorithm 4 comes from the i.i.d. random selection variable i_t^k only. The statistical analysis to relate the SAA problem in (4.1) and the original problem in (1.1) has been extensively studied especially under the stage-wise independence assumption (e.g.

Algorithm 4 Stochastic dual dynamic programming (SDDP)

```

1: Set  $V_t^0(x) = -\infty$ ,  $t = 2, \dots, T$ ,  $\underline{V}_{T+1}^k(x) = 0$ ,  $k \geq 1$ .
2: for  $k = 1, 2, \dots$ , do
3:   for  $t = 1, \dots, T$  do                                 $\triangleright$  Forward phase.
4:     Pick up  $i_t \equiv i_t^k$  from  $\{1, 2, \dots, N_t\}$  uniformly randomly.
5:     Set
6:      $x_t^k \in \operatorname{Argmin}_{x \in X_t(x_{t-1}^k, \tilde{\xi}_{ti_t})} \{F_{ti_t}^{k-1}(x) := H_t(x, \tilde{c}_{ti_t}) + \lambda \underline{V}_{t+1}^{k-1}(x)\}$ .      (5.1)
7:   end for
8:   for  $t = T, T-1, \dots, 2$  do                                 $\triangleright$  Backward phase.
9:     for  $i = 1, \dots, N_t$  do
10:      Set  $\tilde{v}_{ti}^k(x_{t-1}^k)$  and  $(\tilde{v}_{ti}^k)'(x_{t-1}^k)$  according to (4.10) and (4.11).
11:      Update  $\underline{V}_t^k(x)$  according to (4.12) and (4.13).
12:   end for
13: end for

```

[27]). The separation of these two problems allows us to greatly simplify the analysis of SDDP.

Whenever the iteration index k is clear from the context, we use the short-hand notation $i_t \equiv i_t^k$. We also use the notation

$$i_{[k,t]} := \{i_1^1, \dots, i_T^1, i_1^2, \dots, i_T^2, \dots, \dots, i_1^{k-1}, \dots, i_T^{k-1}, i_1^k, \dots, i_t^k\}$$

to denote the sequence of random selection variables generated up to stage t at the k -th iteration. The notions $i_{[k,0]}$ and $i_{[k-1,T]}$ will be used interchangeably. As a filtration is a sequence of sigma-algebras, this statement seems imprecise. It seems to me that $I_{k,t}$ is a sigma-algebra but not a filtration, while $I_{k,t}$ is a filtration. It should be noted that for any iteration $k \geq 1$, we must have $i_1^k = 1$ since the number of scenarios $N_1 = 1$. In other words, i_1^k is always deterministic for any $k \geq 1$.

The complexity analysis of SDDP still relies on the concept of saturation. Let us denote S_t^{k-1} the set of saturated points in stage t , i.e., $S_t^{k-1} := \{x_t : V_{t+1}(x_t) - \underline{V}_{t+1}^j(x_t) \leq \epsilon_t, \text{ for some } j \leq k-1\}$. We still use $x_t^{j_i}$ for some $j_i < k-1$ to denote the closest point to \tilde{x}_{ti}^k from the saturated points S_t^{k-1} , i.e.,

$$x_t^{j_i} \in \operatorname{Argmin}_{s \in S_t^{k-1}} \|s - \tilde{x}_{ti}^k\|, \quad (5.2)$$

$$V_{t+1}(x_t^{j_i}) - \underline{V}_{t+1}^{j_i}(x_t^{j_i}) \leq \epsilon_t. \quad (5.3)$$

In SDDP, we will explore the average distance between \tilde{x}_{ti}^k to the set S_t^{k-1} defined as follows:

$$\tilde{g}_t^k := \frac{1}{N_t} \sum_{i=1}^{N_t} \|\tilde{x}_{ti}^k - x_t^{j_i}\|. \quad (5.4)$$

Note that the search point x_t^k is a function of $i_{[k,t]}$ and hence is also random. \tilde{x}_{ti}^k depends on x_{t-1}^k (see (4.8)) and hence on $i_{[k,t-1]}$. Moreover, the set of saturated points S_t^{k-1} only depends on $i_{[k-1,T]}$ since it is defined in the backward phase of the previous iteration. Hence, \tilde{g}_t^k is measurable w.r.t. $\mathcal{I}_{k,t-1}$, but it is independent of the random selection variable i_t^k for the current stage t at the k -th iteration.

Lemma 8 below summarizes some important properties about \tilde{g}_t^k .

Lemma 8 *Let $\delta_t \in [0, +\infty)$ be given and ϵ_t be defined in (3.24). If $\tilde{g}_t^k \leq \delta_t$, then we have*

$$\frac{1}{N_t} \sum_{i=1}^{N_t} [F_{ti}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{k-1}(\tilde{x}_{ti}^k)] = \frac{\lambda}{N_t} \sum_{i=1}^{N_t} [V_{t+1}(\tilde{x}_{ti}^k) - \underline{V}_{t+1}^{k-1}(\tilde{x}_{ti}^k)] \leq \epsilon_{t-1}. \quad (5.5)$$

Moreover, for $t \geq 2$ we have

$$V_t(x_{t-1}^k) - \underline{V}_t^k(x_{t-1}^k) \leq \epsilon_{t-1}. \quad (5.6)$$

Proof First note the second inequality in (4.24) still holds since it does not depend on the selection of x_t^k . Hence we have

$$F_{ti}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{k-1}(\tilde{x}_{ti}^k) \leq (M_t + \underline{M}_t) \|\tilde{x}_{ti}^k - x_t^{j_i}\| + \lambda \epsilon_t.$$

Summing up the above inequalities, we can see that

$$\begin{aligned} \frac{1}{N_t} \sum_{i=1}^{N_t} [F_{ti}(\tilde{x}_{ti}^k) - \underline{F}_{ti}^{k-1}(\tilde{x}_{ti}^k)] &\leq (M_t + \underline{M}_t) \frac{1}{N_t} \sum_{i=1}^{N_t} \|\tilde{x}_{ti}^k - x_t^{j_i}\| + \lambda \epsilon_t \\ &= (M_t + \underline{M}_t) \tilde{g}_t^k + \lambda \epsilon_t \\ &\leq \epsilon_{t-1}, \end{aligned}$$

which together with the definitions of F_{ti} and \underline{F}_{ti}^{k-1} then imply (5.5). Moreover, (5.6) follows from (4.27) and (5.5). \square

Similar to the previous section, we use

$$g_t^k(x_t^k) := \begin{cases} \min_{s \in S_t^{k-1}} \|s - x_t^k\|, & t < T, \\ 0, & \text{o.w.} \end{cases}$$

to measure the distance between x_t^k and the set of saturated points. Clearly, $g_t^k(x_t^k)$ is a random variable dependent on x_t^k and hence measurable w.r.t. $\mathcal{I}_{k,t}$. We say that x_t^k is ϵ_t -saturated if $V_{t+1}^k(x_t^k) - \underline{V}_{t+1}(x_t^k) \leq \epsilon_t$. Moreover, x_t^k is said to be δ_t -distinguishable if $g_t^k(x_t^k) > \delta_t$.

The quantitates \tilde{g}_t^k and \tilde{g}_{t+1}^k defined in (5.4) provide us a way to check whether x_t^k is δ_t -distinguishable and ϵ_t -saturated. More specifically, If $\tilde{g}_t^k > \delta_t$ for some stage $t < T$ at iteration k , then there must exist an index $i_t^* \equiv i_t^{k,*} \in \{1, \dots, N_t\}$ s.t.

$\|\tilde{x}_{ti_t^*}^k - x_t^{j_{i_t^*}}\| > \delta$ or equivalently $g_t^k(\tilde{x}_{ti_t^*}^k) > \delta_t$ (since otherwise $\tilde{g}_t^k \leq \delta_t$). Note that both \tilde{g}_t^k and i_t^* are measurable w.r.t. $\mathcal{I}_{k,t-1}$ but independent of the i_t^k . Therefore, conditioning on $\mathcal{I}_{k,t-1}$ the probability of having $i_t^k = i_t^*$ is $1/N_t$, and consequently by the law of total probability, $\text{Prob}\{x_t^k = \tilde{x}_{ti_t^*}^k\} = 1/N_t$. Moreover, we can see that the conditional probability of

$$\begin{aligned} \text{Prob}\{g_t^k(x_t^k) > \delta_t | \tilde{g}_t^k > \delta_t\} &= \sum_{i=1}^{N_t} \frac{1}{N_t} \text{Prob}\{g_t^k(\tilde{x}_{ti}^k) > \delta_t | \tilde{g}_t^k > \delta_t\} \\ &\geq \frac{1}{N_t} \text{Prob}\{g_t^k(\tilde{x}_{ti_t^*}^k) > \delta_t | \tilde{g}_t^k > \delta_t\} \\ &= \frac{1}{N_t}. \end{aligned} \quad (5.7)$$

In other words, if $\tilde{g}_t^k > \delta_t$, then with probability at least $1/N_t$, x_t^k will be δ_t -distinguishable. If, in addition, $\tilde{g}_{t+1}^k \leq \delta_{t+1}$, then in view of Lemma 8, we have $V_{t+1}^k(x_t^k) - V_{t+1}(x_t^k) \leq \epsilon_t$ and hence x_t^k will be ϵ_t -saturated.

While EDDP can find at least one new saturated and distinguishable search point in every iteration, SDDP can only guarantee so in probability as shown in the following result. We use the random variable \mathbf{q}^k to denote whether there exists such a point among any stages at iteration k . Clearly, \mathbf{q}^k is measurable w.r.t. $\mathcal{I}_{k,T}$.

Lemma 9 *Assume that $\delta_t \in [0, +\infty)$, $t = 1, \dots, T$, are given. Also let ϵ_t , $t = 0, \dots, T$, be defined in (3.36). The probability of finding a new δ_t -distinguishable and ϵ_t -saturated and search point at the k -iteration of SDDP can be bounded by*

$$\text{Prob}\{\mathbf{q}^k = 1\} \geq \frac{1}{N} (1 - \text{Prob}\{\tilde{g}_i^k \leq \delta_i, i = 1, \dots, T-1\}), \quad (5.8)$$

where

$$\bar{N} := \max\{N_1, \dots, N_T\}. \quad (5.9)$$

Proof First note that ϵ_t in (3.36) is computed according to the recursion $\epsilon_{t-1} = (M_t + \underline{M}_t)\delta_t + \lambda\epsilon_t$ and the assumption $\epsilon_{T-1} = 0$. Next observe that exactly one of the following T cases will happen at the k -th iteration of the SDDP method.

Case 1: $\tilde{g}_t^k \leq \delta_t$, $\forall 1 \leq t \leq T-1$;

Case t , $t = 2, \dots, T-1$: $\tilde{g}_i^k \leq \delta_i$, $\forall t \leq i \leq T-1$, and $\tilde{g}_{t-1}^k > \delta_{t-1}$;

Case T : $\tilde{g}_{T-1}^k > \delta_{T-1}$.

We use the random variable $\mathbf{q}_t^k = 1$ or 0 to denote whether a δ_t -distinguishable and ϵ_t -saturated search point is found for case t .

Let us start with the T -th case. By Lemma 7, x_{T-1}^k in stage $T-1$ will get 0-saturated. Moreover, by (5.7), we have

$$\text{Prob}\{g_{T-1}^k(x_{T-1}^k) > \delta_{T-1} | \tilde{g}_{T-1}^k > \delta_{T-1}\} \geq \frac{1}{N_{T-1}}.$$

Hence, the probability of finding a new 0-saturated and δ_{T-1} -distinguishable search point for the T -th case can be bounded by

$$\begin{aligned}
\text{Prob}\{\mathbf{q}_T^k = 1\} &\geq \text{Prob}\{g_{T-1}^k(x_{T-1}^k) > \delta_{T-1}\} \\
&= \text{Prob}\{g_{T-1}^k(x_{T-1}^k) > \delta_{T-1} | \tilde{g}_{T-1}^k > \delta_{T-1}\} \text{Prob}\{\tilde{g}_{T-1}^k > \delta_{T-1}\} \\
&\geq \frac{1}{N_{T-1}} \text{Prob}\{\tilde{g}_{T-1}^k > \delta_{T-1}\} \\
&\geq \frac{1}{N} \text{Prob}\{\tilde{g}_{T-1}^k > \delta_{T-1}\}.
\end{aligned} \tag{5.10}$$

Now consider the t -th case for $t = 2, \dots, T-2$. By assumption, we have $\tilde{g}_i^k \leq \delta_i, i = t, \dots, T-1$, implying $\tilde{g}_t^k \leq \delta_t$. Hence, by Lemma 8, we have $V_t^k(x_{t-1}^k) - \underline{V}_t(x_{t-1}^k) \leq \epsilon_{t-1}$. In addition, it follows from the assumption $\tilde{g}_{t-1}^k > \delta_{t-1}$ and (5.7) that with probability at least $1/N_{t-1}$, x_{t-1}^k is δ_{t-1} -distinguishable. Therefore, we conclude that

$$\begin{aligned}
&\text{Prob}\{V_t^k(x_{t-1}^k) - \underline{V}_t(x_{t-1}^k) \leq \epsilon_{t-1}, g(x_{t-1}^k) > \delta_{t-1} \\
&\quad | \tilde{g}_{t-1}^k > \delta_{t-1}; \tilde{g}_i^k \leq \delta_i, i = t, \dots, T-1\} \geq \frac{1}{N_{t-1}}
\end{aligned}$$

and hence that

$$\begin{aligned}
\text{Prob}\{\mathbf{q}_t^k = 1\} &\geq \text{Prob}\{V_t^k(x_{t-1}^k) - \underline{V}_t(x_{t-1}^k) \leq \epsilon_{t-1}, g(x_{t-1}^k) > \delta_{t-1}\} \\
&\geq \frac{1}{N_{t-1}} \text{Prob}\{\tilde{g}_{t-1}^k > \delta_{t-1}; \tilde{g}_i^k \leq \delta_i, i = t, \dots, T-1\} \\
&\geq \frac{1}{N} \text{Prob}\{\tilde{g}_{t-1}^k > \delta_{t-1}; \tilde{g}_i^k \leq \delta_i, i = t, \dots, T-1\}
\end{aligned} \tag{5.11}$$

for any $t = 2, \dots, T-2$. Combining (5.10) and (5.11), we conclude that the probability of finding a new ϵ_t -saturated and δ_t -distinguishable search point at iteration k of SDDP can be bounded by

$$\begin{aligned}
\text{Prob}\{\mathbf{q}^k = 1\} &\geq \sum_{t=2}^T \text{Prob}\{\mathbf{q}_t^k = 1\} \\
&\geq \frac{1}{N} \sum_{t=2}^T \text{Prob}\{\tilde{g}_t^k > \delta_t; \tilde{g}_i^k \leq \delta_i, i = t+1, \dots, T-1\} \\
&= \frac{1}{N} (1 - \text{Prob}\{\tilde{g}_i^k \leq \delta_i, i = 1, \dots, T-1\}).
\end{aligned}$$

□

In view of Lemma 9, one of the following three different cases will happen for each SDDP iteration: (a) $\tilde{g}_i^k \leq \delta_i$ for all $i = 1, \dots, T-1$. The probability of this case is denoted by $\text{Prob}\{\tilde{g}_i^k \leq \delta_i, i = 1, \dots, T-1\}$; (b) A new ϵ_t -saturated and δ_t -distinguishable search point will be generated with probability at least

$$\frac{1}{N} (1 - \text{Prob}\{\tilde{g}_i^k \leq \delta_i, i = 1, \dots, T-1\});$$

and (c) none of the above situation will happen, implying that this particular SDDP iteration is not productive.

Observe that if for some iteration k , we have $\tilde{g}_t^k \leq \delta_t$ for all $t = 1, \dots, T-1$. Then by Lemma 8 (with $t=1$), we have

$$F_{11}(x_1^k) - F^* \leq F_{11}(x_1^k) - \underline{F}_{11}^{k-1}(x_1^k) \leq \epsilon_0. \quad (5.12)$$

Moreover, we have

$$\frac{\lambda}{N_t} \sum_{i=1}^{N_t} [V_{t+1}(\tilde{x}_{ti}^k) - \underline{V}_{t+1}^{k-1}(\tilde{x}_{ti}^k)] \leq \epsilon_{t-1}$$

for all $t = 1, \dots, T$. This observation together with the fact that x_t^k is randomly chosen from \tilde{x}_{ti}^k , $i = 1, \dots, N_t$, then imply that the expectation of $V_{t+1}(x_t^k) - \underline{V}_{t+1}^{k-1}(x_t^k)$ conditionally on $i_{[k,t-1]}$:

$$\begin{aligned} \mathbb{E}[V_{t+1}(x_t^k) - \underline{V}_{t+1}^{k-1}(x_t^k) | \mathcal{I}_{k,t-1}] &= \frac{\lambda}{N_t} \sum_{i=1}^{N_t} [V_{t+1}(\tilde{x}_{ti}^k) - \underline{V}_{t+1}^{k-1}(\tilde{x}_{ti}^k)] \\ &\leq \epsilon_{t-1}, \quad t = 1, \dots, T. \end{aligned} \quad (5.13)$$

Similar in spirit to Lemma 4, the following result relates the above notion of saturation to the gap between a stochastic upper bound and lower bound on the optimal value of problem (4.1).

Lemma 10 *Suppose that the relations in (5.13) hold for some iteration $k \geq 1$. Then we have*

$$\sum_{t=1}^T \lambda^{t-1} \mathbb{E}[H_t(x_t^k, \tilde{c}_{ti_t}) | \mathcal{I}_{k,t-1}] - \mathbb{E}[\underline{F}_{11}^{k-1}(x_1^k) | \mathcal{I}_{k-1,T}] \leq \sum_{t=1}^T \lambda^{t-1} \epsilon_{t-1}. \quad (5.14)$$

Proof Note that we have $N_1 = 1$. By the definition of x_1^k in (5.1) and our assumption in (5.13), we have

$$\begin{aligned} &\mathbb{E}[H_1(x_1^k, \tilde{c}_{t1}) + \lambda V_2(x_1^k) - \underline{F}_{11}^{k-1}(x_1^k) | \mathcal{I}_{k-1,T}] \\ &= \mathbb{E}[H_1(x_1^k, \tilde{c}_{t1}) + \lambda V_2(x_1^k) | \mathcal{I}_{k-1,T}] - \mathbb{E}[H_1(x_1^k, \tilde{c}_{t1}) + \lambda \underline{V}_2^{k-1}(x_1^k) | \mathcal{I}_{k-1,T}] \\ &= \lambda \mathbb{E}[V_2(x_1^k) - \underline{V}_2^{k-1}(x_1^k) | \mathcal{I}_{k-1,T}] \\ &\leq \lambda \epsilon_1. \end{aligned} \quad (5.15)$$

Now consider the t -th stage for any $t \geq 2$. By the definition of x_t^k in (5.1), we have

$$\begin{aligned} H_t(x_t^k, \tilde{c}_{ti_t}) + \lambda \underline{V}_{t+1}^{k-1}(x_t^k) &= \min\{H_t(x, \tilde{c}_{ti_t}) + \lambda \underline{V}_{t+1}^{k-1}(x) : x \in X_t(x_{t-1}^k)\} \\ &\leq \min\{H_t(x, \tilde{c}_{ti_t}) + \lambda V_{t+1}(x) : x \in X_t(x_{t-1}^k)\} \\ &= v_{ti_t}(x_{t-1}^k) \end{aligned}$$

for any $t \geq 2$. Taking conditional expectation on both sides of the above inequality and using our assumption $\lambda \mathbb{E}[V_{t+1}(x_t^k) - \underline{V}_{t+1}^{k-1}(x_t^k) | \mathcal{I}_{k,t-1}] \leq \epsilon_{t-1}$, we then have

$$\begin{aligned}\mathbb{E}[H_t(x_t^k, \tilde{c}_{ti_t}) + \lambda V_{t+1}(x_t^k) | \mathcal{I}_{k,t-1}] &\leq \mathbb{E}[v_{ti_t}(x_{t-1}^k) | \mathcal{I}_{k,t-1}] + \epsilon_{t-1} \\ &= \mathbb{E}[V_t(x_{t-1}^k) | \mathcal{I}_{k,t-1}] + \epsilon_{t-1} \\ &= \mathbb{E}[V_t(x_{t-1}^k) | \mathcal{I}_{k,t-2}] + \epsilon_{t-1},\end{aligned}$$

where the first identity follows from the definition of V_t and the selection of i_t , and the second identity follows from the fact that x_{t-1}^k is independent of i_t^k . Multiplying λ^{t-1} to both side of the above inequalities, summing them up with the inequalities in (5.15), and using the fact that $V_{T+1}(x_T^k) = 0$, we have

$$\sum_{t=1}^T \lambda^{t-1} \mathbb{E}[H_t(x_t^k, \tilde{c}_{ti_t}) | \mathcal{I}_{k,t-1}] - \mathbb{E}[\underline{F}_{11}^{k-1}(x_1^k) | \mathcal{I}_{k-1,T}] \leq \sum_{t=1}^T \lambda^{t-1} \epsilon_{t-1}.$$

□

We also need to use the following well-known result for the martingale difference sequence when establishing the iteration complexity of SDDP.

Lemma 11 *Let $\xi_{[t]} \equiv \{\xi_1, \xi_2, \dots, \xi_t\}$ be a sequence of iid random variables, and $\zeta_t = \zeta_t(\xi_{[t]})$ be deterministic Borel functions of $\xi_{[t]}$ such that $\mathbb{E}_{\xi_{[t-1]}}[\zeta_t] = 0$ a.s. and $\mathbb{E}_{\xi_{[t-1]}}[\exp\{\zeta_t^2/\sigma_t^2\}] \leq \exp\{1\}$ a.s., where $\sigma_t > 0$ are deterministic. Then*

$$\forall \lambda \geq 0 : \text{Prob} \left\{ \sum_{t=1}^N \zeta_t > \lambda \sqrt{\sum_{t=1}^N \sigma_t^2} \right\} \leq \exp\{-\lambda^2/3\}. \quad (5.16)$$

and

$$\forall \lambda \geq 0 : \text{Prob} \left\{ \sum_{t=1}^N \zeta_t < -\lambda \sqrt{\sum_{t=1}^N \sigma_t^2} \right\} \leq \exp\{-\lambda^2/3\}. \quad (5.17)$$

Proof The proof of (5.16) can be found, e.g., Lemma 2 in [16]. In addition, (5.17) follows from (5.16) by replacing ζ_t with $-\zeta_t$. □

We are now ready to establish the complexity of SDDP.

Theorem 3 *Suppose that the norm used to define the bound D_t in (4.4) is the l_∞ norm. Also assume that $\delta_t \in [0, +\infty)$ and ϵ_t are defined in (3.36). Let K denote the number of iterations performed by SDDP before it finds a forward path (x_1^k, \dots, x_T^k) defined in (5.1) for problem (4.1) s.t.*

$$F_{11}(x_1^k) - F^* \leq \epsilon_0, \quad (5.18)$$

$$\sum_{t=1}^T \lambda^{t-1} \mathbb{E}[H_t(x_t^k, \tilde{c}_{ti_t}) | \mathcal{I}_{k,t-1}]$$

$$-\mathbb{E}[F_{11}^{k-1}(x_1^k)|\mathcal{I}_{k-1,T}] \leq \sum_{t=1}^T \lambda^{t-1} \epsilon_{t-1}. \quad (5.19)$$

Then we have $\mathbb{E}[K] \leq \bar{K}\bar{N} + 2$, where \bar{K} and \bar{N} are defined in (4.31) and (5.9), respectively. In addition, for any $\alpha \geq 1$, we have

$$\text{Prob}\{K \geq \alpha\bar{K}\bar{N} + 1\} \leq \exp\left(-\frac{(\alpha-1)^2\bar{K}^2}{2\alpha\bar{N}}\right). \quad (5.20)$$

Proof First note that if $\tilde{g}_t^k \leq \delta_t$ for all $t = 1, \dots, T-1$, then (5.18) and (5.19) must hold in view of the discussions after Lemma 9 (c.f. (5.12) and (5.13)) and Lemma 10. Therefore, the event $\tilde{g}_t^k \leq \delta_t$ for all $t = 1, \dots, T-1$ will not happen for any $1 \leq k \leq K-1$. In other words, we have $\text{Prob}\{\tilde{g}_t^k \leq \delta_t, t = 1, \dots, T-1\} = 0$ for all $1 \leq k \leq K-1$, which, in view of (5.8), implies that for any $1 \leq k \leq K-1$,

$$\text{Prob}\{\mathbf{q}^k = 1\} \geq \frac{1}{\bar{N}}. \quad (5.21)$$

Moreover, observe that we must have

$$\sum_{k=1}^{K-2} \mathbf{q}^k \leq \bar{K}, \quad (5.22)$$

since otherwise the algorithm has generated totally \bar{K} ϵ_t -saturated and δ_t -distinguishable search points during the first $K-2$ iterations, and thus must terminate at the $K-1$ iterations (i.e., (5.18) and (5.19) must hold due to $\tilde{g}_t^{K-1} \leq \delta_t$ for all $t = 1, \dots, T-1$). Taking expectation on both sides of (5.22), we have

$$\bar{K} \geq \mathbb{E}_K[\mathbb{E}\left[\sum_{k=1}^{K-2} \mathbf{q}^k | K\right]] \geq \mathbb{E}_K\left[\frac{K-2}{\bar{N}}\right] = \frac{\mathbb{E}[K]-2}{\bar{N}},$$

implying that $\mathbb{E}[K] \leq \bar{N}\bar{K} + 2$.

Now we need to bound the probability that the algorithm does not terminate in $\alpha\bar{N}\bar{K} + 1$ iterations for $\alpha \geq 1$. Observe that

$$\text{Prob}\{K \geq \alpha\bar{N}\bar{K} + 1\} \leq \text{Prob}\left\{\sum_{k=1}^{\alpha\bar{N}\bar{K}} \mathbf{q}^k < \bar{K}\right\}, \quad (5.23)$$

since $K \geq \alpha\bar{N}\bar{K} + 1$ must imply that $\sum_{k=1}^{\alpha\bar{N}\bar{K}} \mathbf{q}^k < \bar{K}$. Note that $\mathbf{q}^k - \mathbb{E}[\mathbf{q}^k]$ is a martingale-difference sequence, and $\mathbb{E}[\exp((\mathbf{q}^k)^2)] \leq 1$. Hence we have

$$\begin{aligned} & \text{Prob}\left\{\sum_{k=1}^{\alpha\bar{N}\bar{K}} \mathbf{q}^k < \alpha\bar{K} - \lambda\sqrt{\alpha\bar{N}\bar{K}}\right\} \\ & \leq \text{Prob}\left\{\sum_{k=1}^{\alpha\bar{N}\bar{K}} \mathbf{q}^k \leq \sum_{k=1}^{\alpha\bar{N}\bar{K}} \mathbb{E}[\mathbf{q}^k] - \lambda\sqrt{\alpha\bar{N}\bar{K}}\right\} \end{aligned}$$

$$\leq \exp(-\lambda^2/2), \forall \lambda > 0, \quad (5.24)$$

where the first inequality follows from the fact that $\mathbb{E}[\mathbf{q}^k] \geq 1/\bar{N}$, $k = 1, \dots, \alpha\bar{N}\bar{K}$, and thus $\sum_{k=1}^{\alpha\bar{N}\bar{K}} \mathbb{E}[\mathbf{q}^k] \geq \alpha\bar{K}$, and the second inequality follows from Lemma 11. Setting

$$\lambda = \frac{(\alpha-1)\bar{K}}{\sqrt{\alpha\bar{N}}}$$

in the above relation, we then conclude that

$$\text{Prob} \left\{ \sum_{k=1}^{\alpha\bar{N}\bar{K}} \mathbf{q}^k < \bar{K} \right\} \leq \exp \left(-\frac{(\alpha-1)^2\bar{K}^2}{2\alpha\bar{N}} \right). \quad (5.25)$$

Combining (5.23) and (5.25), we then conclude that

$$\text{Prob}\{K \geq \alpha\bar{N}\bar{K} + 1\} \leq \exp \left(-\frac{(\alpha-1)^2\bar{K}^2}{2\alpha\bar{N}} \right), \forall \alpha \geq 1.$$

□

We have the following immediate consequence of Theorem 3.

Corollary 1 Suppose that $n_t \leq n$, $D_t \leq D$, $\max\{M_t, \underline{M}_t\} \leq M$ and $\delta_t = \epsilon$ for all $t = 1, \dots, T$. Let K denote the number of iterations performed by the SDDP method before it finds a forward path (x_1^k, \dots, x_T^k) of problem (4.1) s.t.

$$F_{11}(x_1^k) - F^* \leq 2M \min\left\{\frac{1}{1-\lambda}, T-1\right\} \epsilon, \quad (5.26)$$

$$\begin{aligned} & \sum_{t=1}^T \lambda^{t-1} \mathbb{E}[H_t(x_t^k, \tilde{c}_{t|t}) | \mathcal{I}_{k,t-1}] - \mathbb{E}[F_{11}^{k-1}(x_1^k) | \mathcal{I}_{k-1,T}] \\ & \leq 2M \min\left\{\frac{1}{(1-\lambda)^2}, \frac{T(T-1)}{2}\right\} \epsilon. \end{aligned} \quad (5.27)$$

Then we have $\mathbb{E}[K] \leq \bar{K}_\epsilon \bar{N} + 2$, where \bar{K}_ϵ and \bar{N} is defined in (4.33) and (5.9), respectively. In addition, for any $\alpha \geq 1$, we have

$$\text{Prob}\{K \geq \alpha\bar{K}_\epsilon \bar{N} + 1\} \leq \exp \left(-\frac{(\alpha-1)^2\bar{K}_\epsilon^2}{2\alpha\bar{N}} \right).$$

Proof The relations in (5.26) and (5.27) follow by using the bound (3.46) for ϵ_0 in (5.18) and by using the bound (3.47) for $\sum_{t=1}^T \epsilon_{t-1}$ in (5.19), respectively. Moreover, the bounds on $\mathbb{E}[K]$ and $\text{Prob}\{K \geq \alpha\bar{K}_\epsilon \bar{N} + 1\}$ directly follows from Theorem 3 by replacing \bar{K} with \bar{K}_ϵ . □

We now add a few remarks about the results obtained in Theorem 3 and Corollary 1. Firstly, since SDDP is a randomized algorithm, we provide bounds on the expected number of iterations required to find an approximate solution of problem (4.1). We

also show that the probability of having large deviations from these expected bounds for SDDP decays exponentially fast. Secondly, the complexity bounds for the SDDP method is \bar{N} times worse than those in Theorem 2 for the EDDP method, even though the dependence on other parameters, including T , n and ϵ , remains the same. Thirdly, similar to DDP and EDDP, the complexity of SDDP actually depends the dimension of the effective feasible region $\tilde{\mathcal{X}}_t$ in (4.31), which can be smaller than n_t .

As shown in Theorem 3 and Corollary 1, we can show the convergence of the gap between a stochastic upper bound on $F_{11}(x_1^k)$, given by $\sum_{t=1}^T \lambda^{t-1} H_t(x_t^k, \tilde{c}_{t i_t})$, and the lower bound $F_{11}^{k-1}(x_1^k)$, generated by the SDDP method. In order to obtain a statistically more reliable upper bound, we can run the forward phase $L \geq 1$ times in each iteration. In particular, we can replace the forward phase in Algorithm 4 with the one shown in Algorithm 5. We can then compute the average and estimated standard deviation of ub_k over these L runs of the forward phase.

Algorithm 5 Forward phase with upper bound estimation

```

1: for  $l = 1, \dots, L$  do ▷ Forward phase.
2:   Set  $\tilde{F}_l = 0$ .
3:   for  $t = 1, \dots, T$  do
4:     Pick up  $i_t$  from  $\{1, 2, \dots, N_t\}$  uniformly randomly.
5:     Set  $x_t^k$  according to (5.1) and  $\tilde{F}_l = \tilde{F}_l + \lambda^{t-1} H_t(x_t^k, \tilde{c}_{t i_t})$ .
6:   end for
7:   Set  $\text{ub}_k = \text{ub}_k + \tilde{F}_l$ .
8: end for
9: Set  $\text{ub}_k = \text{ub}_k / L$ .

```

It should be noted, however, that the convergence of the SDDP method only requires $L = 1$. To choose $L > 1$ helps to properly terminate the algorithm by providing a statistically more accurate upper bound. Moreover, since each run of the forward phase will generate a forward path, we can use these L forward paths to run the backward phases in parallel to accelerate the convergence of SDDP. Following a similar analysis to the basic version of SDDP, we can show that the number of iterations required by the above variant of SDDP will be L times smaller than the one for Algorithm 4, but each iteration is computationally more expensive or requires more computing resources for parallel processing.

6 Conclusion

In this paper, we establish the complexity of a few cutting plane algorithms, including DDP, EDDP and SDDP, for solving dynamic convex optimization problems. These methods build up piecewise linear functions to approximate the value functions through the backward phase and generate feasible policies in the forward phase by utilizing these cutting plane models. For the first time in the literature, we establish the total number of iterations required to run these forward and backward phases in order to compute a certain accurate solution. Our results reveal that these methods have a mild dependence on the number of stages T .

It is worth noting that in our current analysis we assume that all the subproblems in the forward and backward phases are solved exactly. However, we can possibly extend the basic analysis to the case when these subproblems are solved inexactly as long as the errors are small enough. Moreover, we did not make any assumptions on how the subproblems are solved. As a result, it is possible to extend our complexity results to multi-stage stochastic binary (or integer) programming problems (see, e.g., [30]). In addition, the major analysis for SDDP presented in this paper does not rely on the convexity, but the Lipschitz continuity of the value functions and their lower approximations. Hence, it seems to be possible to adapt our analysis for SDDP-type methods with nonconvex approximations for the value functions [1,23].

We have discussed a few different ways to terminate DDP, EDDP and SDDP. More specifically, DDP can be terminated by calculating the gap between the upper and lower bounds, and EDDP is a variant of SDDP with rigorous termination based on the saturation of search points, whereas SDDP is usually terminated by resorting to statistically valid upper bounds coupled with the lower bounds obtained from the cutting plane models. Recently an important line of research has been developed to design SDDP-like methods with more reliable and efficient termination criterions (see, e.g., [3,7,18]). It will be interesting to study the complexity of these new methods in the future.

References

1. Ahmed, S., Cabral, F.G., Costa, B.F.P.D.: Stochastic lipschitz dynamic programming (2019)
2. Bao, H., Zhou, Z., Kotsalis, G., Lan, G., Tong, Z.: Lignin valorization process control under feedstock uncertainty through a dynamic stochastic programming approach. *React. Chem. Eng.* **4**, 1740–1747 (2019)
3. Baucke, R., Downward, A., Zakeri, G.: A deterministic algorithm for solving multistage stochastic programming problems. Technical report, The University of Auckland, 70 Symonds Street, Grafton, Auckland, July 2017 (2017)
4. Birge, J.R.: Decomposition and partitioning methods for multistage stochastic linear programs. *Oper. Res.* **33**(5), 989–1007 (1985)
5. Birge, J.R., Louveaux, F.V.: *Introduction to Stochastic Programming*. Springer, New York (1997)
6. Donohue, C.J., Birge, J.R.: The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithm. Oper. Res.* **1**(1), 20 (2006)
7. Georghiou, A., Tsoukalas, A., Wiesemann, W.: Robust dual dynamic programming. *Oper. Res.* **67**(3), 813–830 (2019)
8. Girardeau, P., Leclerc, V., Philpott, A.B.: On the convergence of decomposition methods for multistage stochastic convex programs. *Math. Oper. Res.* **40**, 130–145 (2015)
9. Guigues, V.: Sddp for some interstage dependent risk-averse problems and application to hydro-thermal planning. *Comput. Optim. Appl.* **57**, 167–203 (2014)
10. Guigues, V.: Inexact cuts in deterministic and stochastic dual dynamic programming applied to linear optimization problems (2018)
11. Higle, J.L., Sen, S.: Stochastic decomposition: an algorithm for two-stage linear programs with recourse. *Math. Oper. Res.* **16**, 650–669 (1991)
12. Hinsberger, M., Philpott, A.B.: Resa: a method for solving multistage stochastic linear programs. *J. Appl. Oper. Res.* **6**(1), 2–15 (2014)
13. Kelley, J.E.: The cutting plane method for solving convex programs. *J. SIAM* **8**, 703–712 (1960)
14. Kozmík, V., Morton, D.P.: Evaluating policies in risk-averse multi-stage stochastic programming. *Math. Program.* **152**(1–2), 275–300 (2015)
15. Lan, G.: *First-Order and Stochastic Optimization Methods for Machine Learning*. Springer, Basel (2020)

16. Lan, G., Nemirovski, A.S., Shapiro, A.: Validation analysis of mirror descent stochastic approximation method. *Math. Program.* **134**, 425–458 (2012)
17. Lan, G., Zhou, Z.: Dynamic stochastic approximation for multi-stage stochastic optimization. Manuscript, Georgia Institute of Technology, 2017. Mathematical Programming, under minor revision (2017)
18. Leclère, V., Carpentier, P., Chancelier, J.P., Lenoir, A., Pacaud, F.: Exact converging bounds for stochastic dual dynamic programming via fenchel duality. *SIAM J. Optim.* **30**(2), 1223–1250 (2020)
19. Linowsky, K., Philpott, A.B.: On the convergence of sampling-based decomposition algorithms for multistage stochastic programs. *J. Optim. Theory Appl.* **125**, 349–366 (2005)
20. Nesterov, Y.E.: *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, Norwell, MA (2004)
21. Pereira, M., Pinto, L.: Multi-stage stochastic optimization applied to energy planning. *Math. Program.* **52**(1–3), 359–375 (1991)
22. Philpott, A., Matos, Vd, Finardi, E.: On solving multistage stochastic programs with coherent risk measures. *Oper. Res.* **61**, 957–970 (2013)
23. Philpott, A., Wahid, F., Bonnans, F.: Midas: A mixed integer dynamic approximation scheme, 2016. PhD thesis, Inria Saclay Ile de France (2016)
24. Tyrrell Rockafellar, R., Wets, Roger J.-B.: Scenarios and policy aggregation in optimization under uncertainty. *Math. Oper. Res.* **16**(1), 119–147 (1991)
25. Ruszczyński, A.: Decomposition methods. In: Ruszczyński, A., Shapiro, A. (eds.) *Stochastic Programming*, pp. 141–211. Elsevier, Amsterdam (2003)
26. Shapiro, A.: On complexity of multistage stochastic programs. *Oper. Res. Lett.* **34**, 1–8 (2006)
27. Shapiro, A.: Analysis of stochastic dual dynamic programming method. *Eur. J. Oper. Res.* **209**, 63–72 (2011)
28. Shapiro, A., Dentcheva, D., Ruszczyński, A.: *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia (2009)
29. Shapiro, A., Nemirovski, A.: On complexity of stochastic programming problems. E-print available at: <http://www.optimization-online.org> (2004)
30. Zou, J., Ahmed, S., Sun, X.A.: Stochastic dual dynamic integer programming. *Math. Program.* **175**(1–2), 461–502 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.