Distributionally Robust Control with Constraints on Linear Unidimensional Projections

Alexandros E. Tzikas,[1] Lukas Fiechtner,[2] Arec Jamgochian,[1,3] and Mykel J. Kochenderfer[1]

Abstract— **Distributionally robust control is a well-studied framework for optimal decision making under uncertainty, with the objective of minimizing an expected cost function over control actions, assuming the most adverse probability distribution from an ambiguity set. We consider an interpretable and expressive class of ambiguity sets defined by constraints on the expected value of functions of one-dimensional linear projections of the uncertain parameters. Prior work has shown that, under conditions, problems in this class can be reformulated as finite convex problems. In this work, we propose two iterative methods that can be used to approximately solve problems of this class in the general case. The first is an approximate algorithm based on best-response dynamics. The second is an approximate method that first reformulates the problem as a semi-infinite program and then solves a relaxation. We apply our methods to portfolio construction and trajectory planning scenarios.**

I. INTRODUCTION

In this work, we consider distributionally robust control (DRC), where the unknown parameters of the system follow an unknown probability distribution. Although the distribution is not known, we assume that it belongs in a known set of probability distributions, called the ambiguity set. This formulation is suited for applications where the probability distribution of the parameters cannot be easily determined or can change over time. Examples include constructing a portfolio without knowledge of the future asset price movements or fixing the energy production of a power plant before the demands are realized [1].

In contrast to DRC, classical control under uncertainty often assumes a known probability distribution for the uncertain parameters of the system. For example, in stochastic optimization, empirical risk minimization (ERM) optimizes the expectation of an objective function l, which estimates costs given parameters x, distributed according to a known distribution $\mathbb{P}$, and design inputs u in set $\mathcal{U}$, with $u^*_{\text{ERM}} = \arg\min_{u \in \mathcal{U}} \mathbb{E}_{x \sim \mathbb{P}}[l(u, x)]$ [2]. In practice, assuming a known

probability distribution for the parameters can be fragile; if the true distribution deviates from the assumed model, the solution's performance can deteriorate severely. Robust control addresses this by optimizing against the worst-case parameter realization within a prespecified set $\mathcal{X}$ with $u^*_{\text{robust}} = \arg\min_{u \in \mathcal{U}} \max_{x \in \mathcal{X}} l(u, x)$ [3], [4]. Unfortunately, specifying a complete and accurate parameter set $\mathcal{X}$ a priori can be impractical. Additionally, worst-case optimization can be overly pessimistic against the real parameters.

DRC bridges the gap between stochastic and robust control by considering uncertainty in the probability distribution of the parameters [1]. Rather than assuming a known distribution or set for the parameters, DRC instead assumes that the parameters follow a distribution that is contained in an ambiguity set $\mathcal{A}$ of distributions. This distributional uncertainty set is often chosen to be a neighborhood around a nominal (e.g. empirical) distribution $\hat{\mathbb{P}}$. This yields a minimax problem where the decision maker chooses a decision input that minimizes the expected cost, assuming a worst-case distribution $\mathbb{P}$:

$$u^*_{\text{DRO}} = \arg\min_{u \in \mathcal{U}} \max_{\mathbb{P} \in \mathcal{A}} \mathbb{E}_{x \sim \mathbb{P}}[l(u, x)]. \quad (1)$$

By optimizing against the worst-case distribution, DRC provides distributional robustness without being overly pessimistic or prescriptive about uncertainty [5], [6].

Solution methods for DRC typically reformulate the problem as a semi-infinite program satisfying constraints over all members of $\mathcal{A}$. They then dualize the problem [7], [8], or iteratively relax the constraints into representative atomic subsets of $\mathcal{A}$ [9], [10]. Typical choices for the ambiguity set consider distributions within an ϵ-ball around a nominal distribution $\hat{\mathbb{P}}$, when $\mathcal{A}$ is defined as $\{\mathbb{P} : D(\mathbb{P}, \hat{\mathbb{P}}) \leq \epsilon\}$, for some discrepancy measure D, such as KL divergence [11], [12], other general f-divergences [13], [14], or Wasserstein distances [15], [16]. Efficient solutions to DRC have been considered for many ambiguity sets [5], [6]. A common challenge in the choice of the ambiguity set that remains, however, is the selection and interoperability of hyperparameters (e.g. the discrepancy radius ϵ), which can significantly affect both robustness and conservatism.

In this work, we identify a specific class of problems in DRC, where the constraints on the probability distribution of the parameters involve expectations of functions of one-dimensional linear projections of the unknown parameters. We show that this problem class is expressive and the constraint set intuitive enough to be determined in practical applications. This simplifies the process of selecting the hyperparameters for the ambiguity set. Prior work has shown

that if strong duality holds for a problem in this class, then it can be reformulated as a finite convex problem, which can be easily solved under mild conditions (e.g., when the conjugate function can be easily evaluated) [1]. *We propose two iterative solution methods to approximately solve the problems of this class in the general case. It turns out that, under conditions, these methods involve problems that can be solved using convex optimization [2] or related heuristics [17].*

This paper is organized as follows. In section II, we introduce the problem class of interest and highlight a few important problems within. We then introduce our two proposed solution methods. In section III, we propose a simple approach to solve problems in this class using best-response dynamics. In section IV, we propose a second solution method that amounts to iteratively solving a semi-infinite program and is based on duality theory and cutting-set algorithms. Next, we present our results and the paper concludes with a brief discussion.

II. PROBLEM STATEMENT

The problem class we consider consists of optimization problems of the form

$$
\min_{u \in \mathcal{U}} \max_{\mathbb{P} \in \mathcal{S}} \mathbb{E}_{x \sim \mathbb{P}}\left[l(u, x)\right]
$$
$$
\text{s.t. } \mathbb{E}_{x \sim \mathbb{P}}\left[g_i(q_i^\top x)\right] \leq 0 \text{ for all } i, \tag{2}
$$

where the input u is constrained to $\mathcal{U}$ and the distribution of the random variable x, $\mathbb{P}$, must belong in $\mathcal{S}$, where $\mathcal{S}$ is the set of probability distributions with support $\mathcal{X}$. The probability distribution $\mathbb{P}$ is further constrained by a set of inequalities. Each inequality constrains the expectation of a function of a one-dimensional linear projection of x. We call this class of problems *DRC-1DP* (DRC with 1-Dimensional Projections).

We next show that the class DRC-1DP admits useful optimization objectives and interpretable and practical constraints. Consider the problem

$$
\min_{u \in \mathcal{U}} \max_{\mathbb{P} \in \mathcal{S}} \mathbb{E}_{x \sim \mathbb{P}}\left[l(u, x)\right]
$$
$$
\text{s.t. } \left|\mathbb{E}_{x \sim \mathbb{P}}\left[g_i(q_i^\top x)\right] - p_i\right| \leq \epsilon_i \text{ for all } i \tag{3}
$$

that belongs in DRC-1DP.

Some common choices for the objective l include bilinear or convex-concave functions. In the bilinear case, $l(u, x)$ is linear in both u and x. This objective has received much attention in the game theory literature [18]. In the convex-concave case, $l(u, x)$ is convex in u and concave in x and the minmax theorem [19] holds: $\min_u \max_x l(u, x) = \max_x \min_u l(u, x)$.

The ambiguity set of problem (3) consists of all probability distributions with support $\mathcal{X}$ that have an expected value of $g_i(q_i^\top x)$ centered around p_i, for all i. To demonstrate the interpretability of the ambiguity set of problem (3), we consider specific cases for g_i below.

- If $g_i(z) = \mathbb{1}\{z \geq b_i\}$, then $\mathbb{E}_{x \sim \mathbb{P}}\left[g_i(q_i^\top x)\right] = \mathbb{P}\{q_i^\top x - b_i \geq 0\}$, and we are constraining the probability content on a selected half-space defined by (q_i, b_i).

- If $g_i(z) = z$, then $\mathbb{E}_{x \sim \mathbb{P}}\left[g_i(q_i^\top x)\right] = \mathbb{E}_{x \sim \mathbb{P}}\left[q_i^\top x\right]$, and we are constraining the expected value of the one-dimensional projection $q_i^\top x$.
- If $g_i(z) = z^p$, then $\mathbb{E}_{x \sim \mathbb{P}}\left[g_i(q_i^\top x)\right] = \mathbb{E}_{x \sim \mathbb{P}}\left[(q_i^\top x)^p\right]$, and we are constraining the p-th central moment of the one-dimensional random variable projection $q_i^\top x$.
- By combining constraints for $g_i(z) = z^2$ and $g_j(z) = z$ with $q_i = q_j$, we are effectively constraining the variance of $q_i^\top x$.

Constraints of the form $\left|\mathbb{E}_{x \sim \mathbb{P}}\left[g_i(q_i^\top x)\right] - p_i\right| \leq \epsilon_i$ for functions g_i in the list above characterize properties of the random vector x that are easily observable and interpretable in practice. For example, if we know that the coordinate j of vector x, x_j, is positive, we can let $g_i(z) = \mathbb{1}\{z \geq 0\}$, q_i be the unit vector in the jth direction, $p_i = 1$, and ϵ_i be very small. The presence of the absolute value in the constraints of (3) and the ability to include higher moment constraints means that (3) does not belong in the class of problems considered in prior work [7].

III. BEST RESPONSE ALGORITHM

An approximation of the main problem (2) is

$$
\min_{u \in \mathcal{U}} \max_{x^{(1)}, \ldots, x^{(n)} \in \mathcal{X}} \frac{1}{n} \sum_{k=1}^n l(u, x^{(k)})
$$
$$
\text{s.t. } \frac{1}{n} \sum_{k=1}^n g_i(q_i^\top x^{(k)}) \leq 0 \text{ for all } i, \tag{4}
$$

where the set of points $\{x^{(1)}, \ldots, x^{(n)}\}$ defines a discrete probability distribution $\mathbb{P}$, with each point assumed equiprobable. In this case, the expectations simplify to sums of n terms, where each term corresponds to a point in $\{x^{(1)}, \ldots, x^{(n)}\}$. Solving problem (4) amounts to choosing the best input under the worst placement of n points in $\mathcal{X}$.

We propose to solve problem (4) using an iterative scheme, as described in algorithm 1. The algorithm iterates over minimizing the cost under a fixed discrete distribution for x and then refining the discrete distribution of equiprobable points in a way that maximizes the expected cost. In other words, we choose a discrete distribution for x in order to maximize cost and we choose the input u in order to minimize the cost. This is an instance of the best-response algorithm, which is common in game-theoretic settings [20].

The performance of algorithm 1 depends on the number of samples n. This needs to be sufficiently large to allow for expressivity and feasibility of the constraints in (6). Finally, we note that there are instances where best-response dynamics do not converge [21].

IV. APPROACH USING SEMI-INFINITE PROGRAMMING

We show how to reformulate problem (2) into a semi-infinite program (SIP) and propose an approximate algorithm to solve the latter.

Algorithm 1 Best-Response Method

Require: discrete set $\hat{\mathcal{X}} = \{\hat{x}^{(1)}, \ldots, \hat{x}^{(n)}\} \subseteq \mathcal{X}$ of samples
1: **repeat**
2:     Solve

$$\min_{u \in \mathcal{U}} \frac{1}{n} \sum_{k=1}^{n} l(u, \hat{x}^{(k)}) \tag{5}$$

   and obtain the optimal control $\hat{u}^*$
3:     Solve

$$\max_{x^{(1)}, \ldots, x^{(n)} \in \mathcal{X}} \frac{1}{n} \sum_{k=1}^{n} l(\hat{u}^*, x^{(k)})$$
$$\text{s.t.} \quad \frac{1}{n} \sum_{k=1}^{n} g_i(q_i^\top x^{(k)}) \leq 0 \text{ for all } i \tag{6}$$

   and assign the optimal variable values $x^{(1),*}, \ldots, x^{(n),*}$
   to be the points $\hat{x}^{(1)}, \ldots, \hat{x}^{(n)}$.
4: **until** convergence
5: **return** the optimal control $\hat{u}^*$

A. Semi-Infinite Programming Formulation

Suppose u is fixed. Then, the inner optimization problem of problem (2)

$$\max_{\mathbb{P} \in \mathcal{S}} \mathbb{E}_{x \sim \mathbb{P}} [l(u, x)]$$
$$\text{s.t.} \ \mathbb{E}_{x \sim \mathbb{P}} [g_i(q_i^\top x)] \leq 0 \text{ for all } i \tag{7}$$

can be equivalently written as

$$\max_{\mathbb{P} \in \mathcal{S}} \min_{\lambda \geq 0} \ \mathbb{E}_{x \sim \mathbb{P}} [l(u, x)] - \sum_i \lambda_i \mathbb{E}_{x \sim \mathbb{P}} [g_i(q_i^\top x)]. \tag{8}$$

Assuming strong duality holds, then problem (8) is equivalent to

$$\min_{\lambda \geq 0} \ \max_{\mathbb{P} \in \mathcal{S}} \mathbb{E}_{x \sim \mathbb{P}} \left[l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) \right]. \tag{9}$$

Under technical conditions, strong duality holds between (8) and (9) [1, Theorem 11].

We can write problem (9) in epigraph form

$$\min_{\lambda \geq 0, t} \quad t$$
$$\text{s.t.} \ \max_{\mathbb{P} \in \mathcal{S}} \mathbb{E}_{x \sim \mathbb{P}} \left[l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) \right] \leq t \tag{10}$$

and observe that

$$\max_{\mathbb{P} \in \mathcal{S}} \mathbb{E}_{x \sim \mathbb{P}} \left[l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) \right] - t \leq 0 \Leftrightarrow$$
$$\max_{x \in \mathcal{X}} \ l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) - t \leq 0. \tag{11}$$

The left implication is obvious. The right implication can be shown by contradiction. Suppose the top condition holds and there exists an $x \in \mathcal{X}$ such that $l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) - t > 0$. Then there exists a probability distribution with support $\mathcal{X}$ that only associates probability mass with the values of x such that $l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) - t > 0$. Therefore, the

expectation under this distribution will be positive and the top condition would not hold. This is a contradiction.

Therefore, problem (9) is equivalent to

$$\min_{\lambda \geq 0, t} \quad t$$
$$\text{s.t.} \ \max_{x \in \mathcal{X}} \ l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) - t \leq 0. \tag{12}$$

Reiterating, under strong duality, (12) is in turn equivalent to (7). We can now include the minimization over u to obtain

$$\min_{u \in \mathcal{U}, \lambda \geq 0, t} \quad t$$
$$\text{s.t.} \quad l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) - t \leq 0 \ \forall \ x \in \mathcal{X}, \tag{13}$$

which is equivalent to (2). This is a semi-infinite program [22].

In the general case, strong duality between (2) and (13) does not hold. However, by weak duality, (13) always constitutes an upper bound of (2). Thus, in this paper we solve (13) as a (potentially conservative) surrogate for (2).

1) Semi-Infinite Program for Problem (3)*:* We can write problem (3) as

$$\min_{u \in \mathcal{U}} \max_{\mathbb{P} \in \mathcal{S}} \mathbb{E}_{x \sim \mathbb{P}} [l(u, x)]$$
$$\text{s.t.} \ \mathbb{E}_{x \sim \mathbb{P}} [g_i(q_i^\top x)] - p_i \leq \epsilon_i, \text{ for all } i \tag{14}$$
$$p_i - \mathbb{E}_{x \sim \mathbb{P}} [g_i(q_i^\top x)] \leq \epsilon_i, \text{ for all } i.$$

This implies that we can write (13) as

$$\min_{u \in \mathcal{U}, \lambda^{(1)} \geq 0, \lambda^{(2)} \geq 0, t} \quad t$$
$$\text{s.t.} \ \max_{x \in \mathcal{X}} \ l(u, x) - \sum_i (\lambda_i^{(1)} - \lambda_i^{(2)}) g_i(q_i^\top x) +$$
$$\sum_i (\lambda_i^{(1)} - \lambda_i^{(2)}) p_i + \sum_i (\lambda_i^{(1)} + \lambda_i^{(2)}) \epsilon_i - t \leq 0. \tag{15}$$

By setting $\alpha = \lambda^{(1)} + \lambda^{(2)}$ and $\beta = \lambda^{(1)} - \lambda^{(2)}$, we obtain the equivalent problem

$$\min_{u \in \mathcal{U}, \alpha \geq 0, \beta, t} \quad t$$
$$\text{s.t.} \ \max_{x \in \mathcal{X}} \ l(u, x) - \sum_i \beta_i g_i(q_i^\top x) + \sum_i \beta_i p_i +$$
$$\sum_i \alpha_i \epsilon_i - t \leq 0 \tag{16}$$
$$\alpha + \beta \geq 0.$$

B. Optimization Algorithm

We propose to approximately solve problem (13) using an iterative cutting-set method [22]. Our proposed approach is described in algorithm 2. The algorithm iterates between solving a relaxation of problem (13), where $\mathcal{X}$ is replaced by a finite set $\hat{\mathcal{X}}$, and then making the relaxation tighter. The relaxation is tightened by computing a worst-case value for x, namely $\hat{x}^*$. The termination criterion implies that upon termination, $(\hat{u}^*, \hat{\lambda}^*, \hat{t}^*)$ is a feasible point of (13).

If $l(u, x)$ is convex in u and $\mathcal{U}$ is a convex set, then we can easily solve problem (17) with convex optimization algorithms [2], [23]. Further, if $l(u, x)$ is concave in x, $\mathcal{X}$ is

Algorithm 2 Cutting-Set Method

Require: discrete set $\hat{\mathcal{X}} = \{x^{(1)}, \ldots, x^{(n)}\} \subseteq \mathcal{X}$
1: **repeat**
2:    Solve

$$\min_{u \in \mathcal{U}, \lambda \geq 0, t} \quad t$$
$$\text{s.t.} \quad l(u, x) - \sum_i \lambda_i g_i(q_i^\top x) - t \leq 0 \ \forall x \in \hat{\mathcal{X}}, \quad (17)$$

   i.e., problem (13) after replacing $\mathcal{X}$ with $\hat{\mathcal{X}}$, and obtain the optimal values for the variables: $\hat{u}^*, \hat{\lambda}^*$, and $\hat{t}^*$
3:    Solve

$$\max_{x \in \mathcal{X}} \ l(\hat{u}^*, x) - \sum_i \hat{\lambda}_i^* g_i(q_i^\top x) - \hat{t}^* \quad (18)$$

   to obtain $\hat{x}^*$
4:    Append $\hat{x}^*$ to $\hat{\mathcal{X}}$
5: **until** $l(\hat{u}^*, \hat{x}^*) - \sum_i \hat{\lambda}_i^* g_i(q_i^\top \hat{x}^*) - \hat{t}^* \leq 0$
6: **return** the optimal control $\hat{u}^*$

---

a convex set, and g_i are convex, the same holds for problem (18).

If for fixed x, $l(u, x)$ can be written as a difference of convex functions in u, then problem (17) can be solved approximately using disciplined convex-concave programming (DCCP) [17]. Similarly, we can use DCCP to solve problem (18), if $l(u, x)$ for fixed u and each function g_i can be written as differences of two convex functions in x. If g_i can be written as differences of two convex functions, then $g_i(q_i^\top x)$ can be written as the difference of two convex functions in x.

V. RESULTS

We test our algorithms from section III and section IV in two scenarios: a portfolio construction problem with application in finance and a trajectory planning problem with application in robotics.

In order to judge the performance of the algorithms, we choose problems (19), (21), and (24), where we can determine the optimal control by inspection. We note that problem (19) can be reformulated as a linear program (LP) and solved exactly. However, this does not hold for problem (21) and problem (24). Further, for the last two problems, we cannot use the prior methodology [1] to reformulate them as equivalent finite convex problems, as the necessary conditions are not satisfied.

A. Portfolio Construction

We consider the problem of constructing a portfolio with three assets [24]. The vector of asset returns is denoted $x \in \mathbb{R}^3$ and we suppose $x \in [-1, 1]^3$. The goal is to determine the optimal portfolio weights, $u \in \mathbb{R}^3$, for a long-only portfolio that maximizes expected return, under uncertainty in the probability distribution of the asset returns

x. Mathematically,

$$\min_{u \in \mathcal{U}} \ \max_{\mathbb{P} \in \mathcal{S}} \ \mathbb{E}_{x \sim \mathbb{P}} \left[-u^\top x \right]$$
$$\text{s.t.} \ |\mathbb{E}_{x \sim \mathbb{P}} [x_1] - 0.3| \leq \epsilon_1,$$
$$|\mathbb{E}_{x \sim \mathbb{P}} [x_2] + 0.1| \leq \epsilon_2, \quad (19)$$
$$|\mathbb{E}_{x \sim \mathbb{P}} [x_3] - 0.2| \leq \epsilon_3,$$
$$\|u\|_1 = 1, \ u \geq 0,$$

where there are constraints on the expected return of each asset and a leverage constraint on the long-only weights. This problem is equivalent to the LP:

$$\max_{\|u\|_1 = 1, \ u \geq 0} (0.3 - \epsilon_1) u_1 + (-0.1 - \epsilon_2) u_2 + (0.2 - \epsilon_3) u_3. \quad (20)$$

We discuss different cases for $(\epsilon_1, \epsilon_2, \epsilon_3)$ below. For both algorithm 1 and algorithm 2, the input set $\hat{\mathcal{X}}$ consists of uniformly randomly sampled points in $[-1, 1]^3$. Problem (19) results in convex programs within algorithm 1 and algorithm 2.

We first suppose that $\epsilon_1 = 0.4$, $\epsilon_2 = 0.2$, and $\epsilon_3 = 0.1$. In this instance of problem (19), only asset 3 always has a positive expected return. Under the worst-case distribution for x, assets 1 and 2 will have a negative expected return. Therefore, the optimal weight vector is $(0, 0, 1)$. This is the weight vector obtained using the cutting-set method, as shown in Table I. However, as shown in Table I, the best-response algorithm gets stuck in a cycle between $(0, 0, 1)$ and $(1, 0, 0)$. This occurs because a best response (among others) to weight vector $(0, 0, 1)$ is the sample $x = (0.2, 0.1, 0.1)$, to which the best response is the weight vector $(1, 0, 0)$.

Second, we consider $\epsilon_1 = 0.2$, $\epsilon_2 = 0.2$, and $\epsilon_3 = 0.1$. In this instance, under the worst-case distribution both assets 1 and 3 have expected return 0.1, while asset 2 has expected return -0.1. Therefore, the optimal weight vectors are of the form $(\tau, 0, 1 - \tau)$ for all $\tau \in [0, 1]$. Both the best-response algorithm and the cutting-set algorithm converge to an optimal weight vector, as shown in Table I.

Third, we consider $\epsilon_1 = \epsilon_2 = \epsilon_3 = 10^{-3}$. Under the worst-case distribution, asset 1 has the best return. Therefore, the optimal weight vector is $(1, 0, 0)$. For small values of $(\epsilon_1, \epsilon_2, \epsilon_3)$, the ambiguity in the distribution of the asset returns decreases. Therefore, we expect it to be more difficult for the best-response algorithm to get trapped in a cycle. We note that in this case both the cutting-set algorithm and the best-response algorithm converge to the optimal weight vector, as shown in Table I.

Finally, we consider the case $\epsilon_1 = \epsilon_2 = \epsilon_3 = 2$. Notice that this implies no actual constraint on the distribution. Any weight vector is identical with respect to the objective of problem (19). The different mechanics of the two presented algorithms lead to different, nevertheless optimal, results, as shown in Table I.

B. Volatility Constraints

We consider the following extension to problem (19):

$$\min_{u \in \mathcal{U}} \ \max_{\mathbb{P} \in \mathcal{S}} \ \mathbb{E}_{x \sim \mathbb{P}} \left[-u^\top x \right]$$

$$\text{s.t.} \ |\mathbb{E}_{x \sim \mathbb{P}} [x_1] - 0.3| \leq 0.4, \ |\mathbb{E}_{x \sim \mathbb{P}} [x_1^2] - 0.34| \leq 0.05,$$
$$|\mathbb{E}_{x \sim \mathbb{P}} [x_2] + 0.1| \leq 0.2, \ |\mathbb{E}_{x \sim \mathbb{P}} [x_2^2] - 0.26| \leq 0.05,$$
$$|\mathbb{E}_{x \sim \mathbb{P}} [x_3] - 0.2| \leq 0.1, \ |\mathbb{E}_{x \sim \mathbb{P}} [x_3^2] - 0.10| \leq 0.05,$$
$$\|u\|_1 = 1, \ u \geq 0.$$
(21)

Constraining the second central moment of each coordinate of the asset return vector corresponds to constraining the volatility of each asset. This problem leads to a non-convex problem (6), for which we use the convex-concave procedure [17]. Because the objective only depends on the expected asset returns, the second moment constraints are irrelevant. This problem is therefore identical to problem (19) for $\epsilon_1 = 0.4$, $\epsilon_2 = 0.2$, and $\epsilon_3 = 0.1$. As shown in Table I, the two algorithms recover the optimal weight vector. Although (21) is equivalent to (19), it is more difficult to solve. In this instance, the best-response algorithm does not get stuck in a cycle, because the extra constraints limit the placement of the samples in (6).

C. Trajectory Planning

We consider the problem of controlling a linear system of the form

$$x_{t+1} = \underbrace{\begin{bmatrix} 0.4 & 1.5 \\ 0 & 0.9 \end{bmatrix}}_{A} x_t + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u_t \ \text{for all } t \geq 0,$$
(22)

where $x_t \in \mathbb{R}^2$ is the state and $u_t \in \mathbb{R}$ the control at time t. Notice that for any $t > 0$, using recursion, we can write

$$x_t = A^t x_0 + \sum_{\tau=0}^{t-1} A^{t-1-\tau} B u_\tau.$$
(23)

We assume that the initial state x_0 is unknown but lies in $\mathcal{X} = [-0.3, 0.3]^2$. We consider the problem of reaching a target position, $z_{\text{goal}} = (0,0)$, at time T, under uncertain initial state x_0:

$$\min_{u_0, \ldots, u_{T-1} \in \mathbb{R}} \ \max_{\mathbb{P} \in \mathcal{S}} \ \mathbb{E}_{x_0 \sim \mathbb{P}} \|z_T - z_{\text{goal}}\|_2$$

$$\text{s.t.} \ |\mathbb{E}_{x_0 \sim \mathbb{P}} [x_1] - 0.1| \leq 0.1,$$
$$|\mathbb{E}_{x_0 \sim \mathbb{P}} [x_2] + 0.1| \leq 0.2,$$
$$u_t \in [-0.1, 0.1] \ \text{for all } t = 0, \ldots, T-1.$$
(24)

By computing the eigenvalues of A, we conclude that the linear dynamical system is asymptotically stable, i.e., the state converges to the origin as time progresses, without any input. Therefore, assuming $T \to \infty$, an optimal open-loop control sequence for (24) is the zero input $\mathbf{0}$.

This is in fact the control input obtained by the two methods for the case of large T ($T = 50$). In Figure 1a, we observe that independently of the initial state, which is randomly sampled in $[-0.3, 0.3]^2$, the state trajectory converges close to the origin. In the case of a smaller T ($T = 10$), the obtained control input is no longer zero.

The shorter horizon requires a nonzero control to force the trajectories close to the origin. In this case, as shown in Figure 1b, it is more difficult to drive all initial states to the origin, with some trajectories ending farther away.

We further consider the case of $T = 50$ and $z_{\text{goal}} = (2,1)$. The optimal control is no longer $\mathbf{0}$, but as shown in Figure 1c, both algorithms drive the state close to $(2,1)$ for randomly sampled initial states in $[-0.3, 0.3]^2$. We include a smaller number of sampled initial states to improve the clarity of the trajectories.

Finally, note that in this set-up, problems (6) and (18) are not convex. We approximately solve them using DCCP. In the trajectory planning set-up, which is the most complicated from the scenarios considered, for the two algorithms, we report the average run-time for a single iteration of their main loop. The implementation is in Python, using the packages CVXPY [23] and DCCP [17]. We run the algorithms on an Apple M1 Pro chip and 32GM of memory. The cutting-set algorithm took $0.29\,\text{s}$ on average for a single iteration of the main loop, while the best-response algorithm required $0.51\,\text{s}$. The cutting-set method is significantly faster, because the size of the involved problems is smaller. Note that the best-response algorithm needs to update the n points in $\mathcal{X}$, by solving (6), in every iteration. In more detail, a convex problem with k variables and m constraints can be solved in roughly $\mathcal{O}\left(\sqrt{m}(k+m)^3\right)$ flops using an interior-point method [2]. Assuming both algorithms involve convex problems, we conclude that the best-response algorithm has a higher complexity. It is therefore reasonable that, for the non-convex case where we apply DCCP, the best-response method also has a higher complexity.

VI. CONCLUSION

We present methods for a class of problems in distributionally robust control where the ambiguity set is defined using 1-dimensional linear projections of the uncertain parameters. This problem class is general and includes problems with interpretable ambiguity sets. Our first method is based on the idea of best-responses from game theory. The second method is based on duality theory and cutting-set methods. We consider simple experimental set-ups and show that the method based on duality theory converges to the optimal control. The method based on best-response dynamics obtained similar convergence in most cases, but it can get trapped in a cycle. Nevertheless, the method based on duality theory also has limitations: it is only exact if strong duality holds and the proposed cutting-set method converges to the solution of problem (13). This method is inferior to the method by Zhen et al. [1], if the necessary conditions that allow for an equivalent finite convex formulation hold, but it can be used to obtain an approximate solution in the general case, where the method by Zhen et al. is not applicable. Future work will deal with the theoretical analysis of both methods that could rely on game theory and fixed point theorems for the best-response method and convex analysis and duality theory for the cutting-set algorithm. Further, an investigation of the cases for which the best-response method gets trapped in a cycle is interesting.

TABLE I: Weight vector solution for the portfolio construction problems (19) and (21) for different selections of $(\epsilon_1, \epsilon_2, \epsilon_3)$. The first entry in each cell corresponds to the weight obtained by the best-response algorithm, while the second entry corresponds to the weight obtained by the cutting-set algorithm. A double-sided arrow $\leftrightarrow$ for the weight vector by the best-response algorithm indicates a cycle.

$(\epsilon_1, \epsilon_2, \epsilon_3)$	Problem (19)				Problem (21)
	$(0.4, 0.2, 0.1)$	$(0.2, 0.2, 0.1)$	$(10^{-3}, 10^{-3}, 10^{-3})$	$(2.0, 2.0, 2.0)$	$(0.4, 0.2, 0.1)$
Asset 1	$0.0 \leftrightarrow 1.0 \mid 0.0$	$0.67 \mid 0.48$	$1.0 \mid 1.0$	$0.7 \mid 0.37$	$0.0 \mid 0.0$
Asset 2	$0.0 \leftrightarrow 0.0 \mid 0.0$	$0.0 \mid 0.0$	$0.0 \mid 0.0$	$0.0 \mid 0.36$	$0.0 \mid 0.0$
Asset 3	$1.0 \leftrightarrow 0.0 \mid 1.0$	$0.33 \mid 0.52$	$0.0 \mid 0.0$	$0.3 \mid 0.27$	$1.0 \mid 1.0$

(a) $T = 50$, $z_{\text{goal}} = (0, 0)$.

(b) $T = 10$, $z_{\text{goal}} = (0, 0)$.

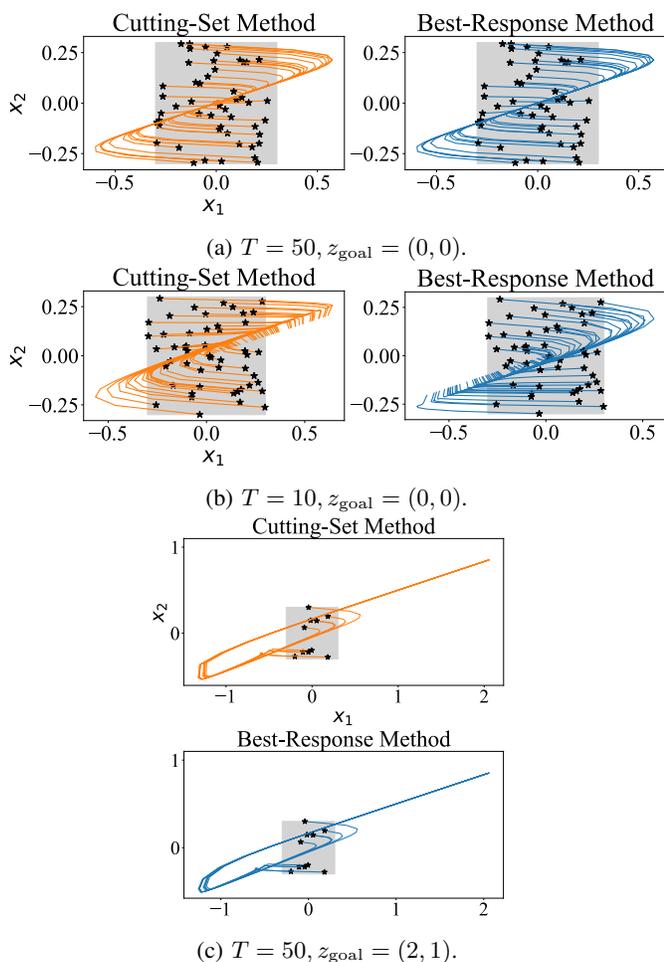(c) $T = 50$, $z_{\text{goal}} = (2, 1)$.

Fig. 1: Trajectories for randomly sampled initial state $x_0 \in [-0.3, 0.3]^2$ and open-loop control as determined by Algorithm 1 and Algorithm 2 for problem (24). The asterisk denotes the initial state for each trajectory.

REFERENCES

[1] J. Zhen, D. Kuhn, and W. Wiesemann, "A Unified Theory of Robust and Distributionally Robust Optimization via the Primal-Worst-Equals-Dual-Best Principle," *Operations Research*, 2023.
[2] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.
[3] A. Ben-Tal and A. Nemirovski, "Robust Optimization–Methodology and Applications," *Mathematical Programming*, vol. 92, pp. 453–480, 2002.
[4] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and Applications of Robust Optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.
[5] H. Rahimian and S. Mehrotra, "Frameworks and Results in Distributionally Robust Optimization," *Open Journal of Mathematical Optimization*, vol. 3, pp. 1–85, 2022.
[6] F. Lin, X. Fang, and Z. Gao, "Distributionally Robust Optimization: A Review on Theory and Applications," *Numerical Algebra, Control and Optimization*, vol. 12, no. 1, pp. 159–212, 2022.
[7] E. Delage and Y. Ye, "Distributionally Robust Optimization under Moment Uncertainty with Application to Data-Driven Problems," *Operations Research*, vol. 58, no. 3, pp. 595–612, 2010.
[8] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally Robust Convex Optimization," *Operations Research*, vol. 62, no. 6, pp. 1358–1376, 2014.
[9] G. Pflug and D. Wozabal, "Ambiguity in Portfolio Selection," *Quantitative Finance*, vol. 7, no. 4, pp. 435–442, 2007.
[10] H. Rahimian, G. Bayraksan, and T. Homem-de Mello, "Identifying effective scenarios in distributionally robust stochastic programs with Total Variation distance," *Mathematical Programming*, vol. 173, no. 1, pp. 393–430, 2019.
[11] A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, "Robust Solutions of Optimization Problems Affected by Uncertain Probabilities," *Management Science*, vol. 59, no. 2, pp. 341–357, 2013.
[12] Z. Hu and L. J. Hong, "Kullback-Leibler Divergence Constrained Distributionally Robust Optimization," *Optimization Online*, vol. 1, no. 2, p. 9, 2013.
[13] H. Namkoong and J. C. Duchi, "Stochastic Gradient Methods for Distributionally Robust Optimization with f-Divergences," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, 2016.
[14] J. C. Duchi and H. Namkoong, "Learning Models with Uniform Performance via Distributionally Robust Optimization," *The Annals of Statistics*, vol. 49, no. 3, pp. 1378–1406, 2021.
[15] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning," in *Operations Research & Management Science in the Age of Analytics*. Informs, 2019, pp. 130–166.
[16] R. Gao, X. Chen, and A. J. Kleywegt, "Wasserstein Distributionally Robust Optimization and Variation Regularization," *Operations Research*, vol. 72, no. 3, pp. 1177–1191, 2024.
[17] X. Shen, S. Diamond, Y. Gu, and S. Boyd, "Disciplined Convex-Concave Programming," *IEEE Conference on Decision and Control*, 2016.
[18] C. J. Li, Y. Yu, N. Loizou, G. Gidel, Y. Ma, N. Le Roux, and M. Jordan, "On the Convergence of Stochastic Extragradient for Bilinear Games using Restarted Iteration Averaging," *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.
[19] D. Goktas and A. Greenwald, "Convex-Concave Min-Max Stackelberg Games," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
[20] M. J. Kochenderfer, T. A. Wheeler, and K. H. Wray, *Algorithms for Decision Making*. MIT Press, 2022.
[21] T. Roughgarden, *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, 2016.
[22] A. Mutapcic and S. Boyd, "Cutting-Set Methods for Robust Convex Optimization with Pessimizing Oracles," *Optimization Methods & Software*, vol. 24, no. 3, pp. 381–406, 2009.
[23] S. Diamond and S. Boyd, "CVXPY: A Python-Embedded Modeling Language for Convex Optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
[24] S. Boyd, K. Johansson, R. Kahn, P. Schiele, and T. Schmelzer, "Markowitz Portfolio Construction at Seventy," *arXiv preprint arXiv:2401.05080*, 2024.