Online First-Order Framework for Robust Convex Optimization

Nam Ho-Nguyen[1] and Fatma Kılınç-Karzan[1]

[1]Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, 15213, USA.

July 20, 2016; revised on October 31, 2016

Abstract

Robust optimization (RO) has emerged as one of the leading paradigms to efficiently model parameter uncertainty. The recent connections between RO and problems in statistics and machine learning domains demand for solving RO problems in ever more larger scale. However, the traditional approaches for solving RO formulations based on building and solving robust counterparts or the iterative approaches utilizing nominal feasibility oracles can be prohibitively expensive and thus significantly hinder the scalability of RO paradigm. In this paper, we present a general and flexible iterative framework to approximately solve robust convex optimization problems that is built on a fully online first-order paradigm. In comparison to the existing literature, a key distinguishing feature of our approach is that it only requires access to first-order oracles that are remarkably cheaper than pessimization or nominal feasibility oracles, while maintaining the same convergence rates. This, in particular, makes our approach much more scalable and hence preferable in large-scale applications, specifically those from machine learning and statistics domains. We also provide new interpretations of existing iterative approaches in our framework and illustrate our framework on robust quadratic programming.

1 Introduction

Robust optimization (RO) is one of the leading modeling paradigms for optimization problems under uncertainty. As opposed to the other approaches, RO seeks a solution that is immunized against *all* possible realizations of uncertain model parameters (noises) from a given uncertainty set. It is widely adopted in practice mainly because of its computational tractability. We refer the reader to the paper by Ben-Tal and Nemirovski [6], the book by Ben-Tal et al. [4] and surveys [8, 9, 11, 14] for a detailed account of RO theory and numerous applications.

Recently, fascinating connections have been established between problems from the statistics and machine learning domains and robust optimization. More precisely, it is demonstrated that RO can be used to achieve desirable statistical properties such as stability, sparsity, and consistency. For example, for linear regression problems, El Ghaoui and Lebret [16] and Xu et al. [35] respectively establish the equivalence of the ridge regression and Lasso to specific RO formulations of unregularized regression problems. Moreover, Xu et al. [34] exhibit similar results in the context of regularizing support vector machines (SVMs), and [34, 35] validate the statistical consistency of methods such as SVM and Lasso via RO methodology. In addition to these RO interpretations of regularization techniques used in statistics and machine learning, robust versions of many problems from these domains are gaining traction. For example, [33] examines robust variants of SVMs and other classification problems, and [2] explores a robust formulation for kernel classification problems. We refer the reader to [14, 5] and references therein for further examples and details on connections between robust optimization and statistics and machine learning.

These recent connections not only highlight the importance of RO methodology but also present algorithmic challenges where the scalability of RO algorithms with problem dimension becomes crucial. The primary method for solving a robust convex optimization problem is to transform it into an equivalent deterministic problem called the *robust counterpart*. Under mild assumptions, this yields a convex and tractable robust counterpart problem (see [4, 11, 3]), which can then be solved using existing convex optimization software and tools. This approach has seen much success in decision making domain. A drawback of this traditional approach is that the reformulated robust counterpart is often not as scalable as the deterministic nominal program. In particular, the robust counterpart can easily belong to a different class of optimization problems as opposed to the underlying original deterministic problem. For example, a linear program (LP) with ellipsoidal uncertainty is equivalent to a convex quadratic program (QP), and similarly, a conic-quadratic program with ellipsoidal uncertainty is equivalent to a semidefinite program (SDP) (see e.g., [4, 11]). It is well-known that convex QPs as opposed to LPs, and SDPs as opposed to convex QPs are much less scalable in practice. This then presents a critical challenge in applying RO methodology in big data applications frequently encountered in machine learning and statistics, where even solving the original deterministic nominal problem to high accuracy is prohibitively time-consuming.

The iterative schemes that alternate between the generation/update of candidate solutions and the realizations of noises offer a convenient remedy to the scalability issues associated with the robust counterpart approach. Thus far, such approaches [26] and [5] have relied on two oracles: (*i*) *solution oracles* to solve instances of extended (or nominal) problems with constraint structures similar to (or the same as) the deterministic problem, and (*ii*) *noise oracles* to generate/update particular realizations of the uncertain parameters. At each iteration of these schemes, both solution and noise oracles are called, and their outputs are used to update the inputs of each other oracle in the next iteration. Because solution oracles rely on a solver of the same class capable of solving the deterministic problem, these iterative approaches circumvent the issue of the robust counterpart approach potentially relying on a different solver. Nevertheless, these iterative approaches still suffer from a serious drawback: the solution oracles in [26, 5] themselves can be expensive as they require solving extended or nominal optimization problems completely. While solving the nominal problem is not as computationally demanding as solving the robust counterpart, the overall procedure depending on repeated calls to such oracles can be prohibitive. In fact, each such call to a solution oracle may endure a significant computational cost, which is at least as much as the computational cost of solving an instance of the deterministic nominal problem. Note that, to ensure scalability, most applications in machine learning and statistics already need to rely on cheap first-order methods for solving deterministic nominal problems.

In this paper, we propose an efficient iterative framework for solving robust convex optimization problems which can rely on, in an *online* fashion, much cheaper *first-order oracles* in place of full solution and noise oracles. In particular, in each iteration, instead of solving a complete optimization problem within the solution and/or noise oracles, we show that simple simultaneous updates on the solution and noise in an online fashion using only first-order information from the deterministic constraint structure is sufficient to solve robust convex optimization problems. Moreover, we show that the number of calls to such online first-order (OFO) oracles is not only at most that of the state-of-the-art iterative approaches utilizing full optimization based oracles for solution and/or noise, but also almost independent of the dimension of the problem. This therefore makes our approach especially attractive for applications in statistics and machine learning domains where it is critical to maintain that the overall approach has both gracious dependence on the dimension of the problem and cheap iterations. We outline our contribution more concretely after discussing the most relevant literature.

Related Work

Thus far, the iterative approaches, which bypass the restrictions of the robust counterparts, work with *extended* nominal problems that belong to the same class as the deterministic nominal one by carefully controlling the constraints included in the formulation corresponding to noise realizations.

For robust binary linear optimization problems with only objective function uncertainty and a polyhedral uncertainty set, Bertsimas and Sim [12] suggest an approach which relies on solving $n + 1$ number of instances of the nominal problem, where n is the dimension of the problem.

For robust convex optimization problems, Calafiore and Campi [13] study a 'constraint sampling' approach based on forming a single extended nominal problem of the same class as the deterministic one via i.i.d. sampling of noise realizations. They show that the optimal solution to this extended nominal problem is robust feasible with high probability where the probability depends on the sampling procedure, the number of samples drawn, and the dimension.

Mutapcic and Boyd [26] follow a 'cutting-plane' type approach where in each iteration, a solution oracle is called to solve an extended nominal problem of the same class as the deterministic problem and a noise oracle, referred to as *pessimization oracle*, is invoked to iteratively expand and refine the extended nominal problem. Given a candidate solution, a pessimization oracle either certifies its feasibility with respect to the robust constraints or returns a new noise realization from the uncertainty set for which the solution is infeasible; then the nominal constraint associated with that particular noise realization is included in the extended problem. This process is repeated until a robust feasible solution is found or the last extended problem is found to be infeasible. In the overall procedure, the number of iterations (or calls to the pessimization oracle) can be exponential in the dimension. Despite this, [26] reports impressive computational results.

Both of the approaches from [13] and [26] pose issues for high-dimensional problems. In [13], as the dimension grows, an extended problem with linearly more nominal constraints is required to ensure the high probability guarantee on finding a good quality solution. In [26] at each iteration, a nominal constraint is added to the extended nominal problem. The theoretical bound on the number of constraints that need to be added is exponential, so the extended problem in [26] can grow to be exponentially large. Moreover, in both cases the extended nominal problem may no longer have certain favorable problem structure of the deterministic nominal problem, such as a network flow structure.

To address these issues, in particular, the issue of solving extended nominal problems that are not only larger-in-size than the deterministic problem but also may lack certain favorable problem structure of the deterministic problem, Ben-Tal et al. [5] introduce a new iterative approach to approximately solve robust feasibility problems via a *nominal feasibility oracle* and running an online learning algorithm to choose noise realizations. Given a particular noise realization, the nominal feasibility oracle solves an instance of the deterministic nominal feasibility problem obtained by simply fixing the noise to the given value. Hence, the problem solved by this oracle has the *same* number of constraints and the *same structure* as the original nominal problem; in particular its size does not grow in each iteration. This is an important distinguishing feature of this approach. The other distinguishing feature is that Ben-Tal et al. [5] replace the pessimization oracle of [26] by employing an online learning algorithm, which simply requires first-order information of the noise from the constraint functions. Moreover, [5] provides a dimension independent bound on the number of iterations (nominal feasibility oracle calls). Because this approach is closely related to our work, we give a detailed summary of it in Section 4.3 and highlight its connections to our work; in fact, we show that it can be seen as a special case of our framework.

We close with a brief summary of the assumptions on the computational requirements of these methods. The constraint sampling approach of [13] requires access to a sampling procedure on the uncertainty sets as well as an oracle capable of solving the extended nominal problem. The cutting plane approach of [26] replaces the sampling procedure of [13] with a noise oracle, namely the pessimization oracle that works with the uncertainty sets but still requires the same type of optimization oracle as a solution oracle to solve the extended problems. Ben-Tal et al. [5] substitute the pessimization oracle with an online learning-based procedure, which requires merely first-order information from the constraint functions and simple projection type operations on the associated uncertainty sets, but it still relies on a solution oracle capable of solving the original nominal problem, which is essentially the same (up to log factors) as the optimization oracles in [13] and [26]. If the deterministic problem admits special structure such as network flows etc., a specific solver can be used in the framework of [5], but this is not possible for [13] and [26].

Summary of Our Contributions

It is possible to view all of these iterative approaches as two iterative processes that run simultaneously and in conjunction with each other to generate/update solutions and noise realizations. This naturally leads to a dynamic game environment where in each round Player 1 chooses a solution and Player 2 chooses a realization of uncertain parameters. In this framework, the policies employed by these players in their decision making determine the nature of the final approach. In the case of [26], Player 1 considers all of the previous noise realizations when making his decision, whereas Player 2 simply reacts to the current solution when choosing the noise. In [5], Player 1 reacts to only the current noise in generating/updating the solution while Player 2 minimizes the regret associated with past solutions in choosing noise.

In this paper, we further analyze this interaction between Player 1 and Player 2, with the aim of deriving a simpler and computationally much less demanding iterative approach to solving RO problems. Our contributions can be summarized as follows.

1. We build a *general and flexible framework* for iteratively solving robust feasibility problems, and demonstrate its flexibility by describing it as a meta-template. By customizing our framework appropriately, we modify the pessimization oracle-based approach of [26] and obtain a much better bound on the number of oracle calls in [26]. We also provide a new interpretation of the nominal feasibility oracle-based approach of [5] as a special case within our framework.

2. When the original deterministic problem admits first-order oracles capable of providing gradient/subgradient information on each constraint function, we demonstrate that *online first-order* (OFO) algorithms can be used to iteratively generate/update solutions and noise realizations simultaneously in an online manner, which also leads to robust feasibility/infeasibility certificates within our framework. In contrast to the approaches of [26] and [5], which rely on full nominal feasibility oracles to generate points, our OFO-based approach only requires simple update rules in each iteration and thus has much lower per-iteration cost. Besides,

our noise oracle generates a realization of the noise in an online learning fashion as was done in [5], and hence it is less expensive than the pessimization oracle of [26].

3. In our framework, the number of iterations (or oracle calls) needed to obtain approximate robust solution or a robust infeasibility certificate is a function of the approximation guarantee ϵ and the complexities of the domains for the solution and the uncertainty set; in particular, our convergence rate is (almost) dimension independent. We also demonstrate that the iteration complexity of our OFO-based approach is at least as good as that of the efficient approach of [5], and better than the exponential complexity of [26]. Overall, our OFO-based approach leads to computational savings over the approach of [5] by a factor as large as $O(1/(\epsilon^2 \log(1/\epsilon)))$ arithmetic operations when the number updates of the solution is smaller than or equal to the number of updates of the noise realization, which is the case in many applications. For further comparisons and discussion, see Section 4.4. In addition, our framework is amenable to exploiting favorable structural properties of the constraint functions such as strong concavity, smoothness, etc., through which better convergence rates can be achieved.

4. Our framework is based on formulating the robust feasibility problem as a convex-nonconcave saddle point (SP) problem, and explicitly analyzing its structure. While convex-concave SP problems are well-studied in the literature, and many efficient first-order algorithms exist for these (see for example [29, 22, 23]), the convex-*nonconcave* SP problem is not as well-studied. To our knowledge, an explicit study of convex-nonconcave SP problems and their relation to RO has not been conducted previously; in this respect, the most closely related work [5] neither provides an explicit connection between robust feasibility and SP problems, nor analyzes their structure explicitly.

To demonstrate the application and effectiveness of our proposed framework, we walk through a detailed example on robust QPs. In particular, for robust QPs, we are able to leverage a recent convex QP-based reformulation of the classical trust region subproblem [20] in order to avoid working with a nonconvex reformulation in a lifted space as in [5, Section 4.2] and relying on a probabilistic follow-the-perturbed-leader type algorithm [5, Section 3.2]. While using such nonconvex techniques will work within our framework, our convex reformulation allows us to work directly in the original space of the variables with a deterministic subgradient-based algorithm while still achieving asymptotically similar iteration complexity guarantees as [5]. Moreover, each iteration of our approach requires only first-order updates where the most expensive operation is the computation of a maximum eigenvector; thus our per-iteration cost is significantly less.

Outline

The rest of the paper is organized as follows. We begin with some notation and preliminaries in Section 2. We introduce our robust feasibility problem and robust feasibility/infeasibility certificates in Section 2.1, convex-concave SP problems in Section 2.2, and briefly summarize important online convex optimization (OCO) tools as well as a useful OFO algorithm in Section 2.3. We formulate the robust feasibility problem as a convex-nonconcave SP problem in Section 3; this formulation and certain bounds associated with its SP gap function form the basis of our general framework for solving robust feasibility problems. In Section 4 we specify an assortment of approaches obtained in our general framework by using different oracles. We examine our OFO-based approach in Section 4.1 by interpreting various terms in our framework in the context of OCO. In Section

4.2, we modify the pessimization oracle-based approach of [26] to obtain an efficient bound on the number of iterations required. In Section 4.3 we show how the nominal feasibility oracle-based approach of [5] fits within our framework. Finally, we discuss the convergence rates and accelerations attainable in our framework and compare our work with the existing approaches in Section 4.4. In Section B we illustrate our OFO-based approach through an example application on robust QPs. We close with a summary of our results and a few compelling further research directions in Section 5. In Appendix A we give an alternative formulation of the robust feasibility problem as a convex-concave SP problem in an extended space, and discuss its advantages and disadvantages over the convex-nonconcave SP formulation.

2 Notation and Preliminaries

Given $a \in \mathbb{R}$, $\operatorname{sign}(a)$ denotes the sign of the number a. For a positive integer $n \in \mathbb{N}$, we let $[n] = \{1, \dots, n\}$ and define $\Delta_n := \{x \in \mathbb{R}^n_+ : \ \sum_{i \in [n]} x_i = 1\}$ to be the standard simplex. Throughout the paper, the superscript, e.g., f^i, u^i, U^i, is used to attribute items to the i-th constraint, whereas the subscript, e.g., x_t, f_t, ϕ_t, is used to attribute items to the t-th iteration. Therefore, we sometimes use u^i, x_t, as well as u^i_t to denote vectors in $\mathbb{R}^n$. We use the notation $\{x_t\}_{t=1}^T$ to denote the collection of items $\{x_1, \dots, x_T\}$. Given a vector $x \in \mathbb{R}^n$, we let $x^{(k)}$ denote its k-th coordinate for $k \in [n]$. One exception we make to this notation is that we always denote the convex combination weights $\theta \in \Delta_T$ with θ_t. For $x \in \mathbb{R}^n$ and $p \in [1, \infty]$, we use $\|x\|_p$ to denote the ℓ_p-norm of x defined as

$$\|x\|_p = \begin{cases} \left(\sum_{i \in [n]} |x^{(i)}|^p\right)^{1/p} & \text{if } p \in [1, \infty) \\ \max_{i \in [n]} |x^{(i)}| & \text{if } p = \infty \end{cases}.$$

Throughout this paper, we use Matlab notation to denote vectors and matrices, i.e., $[x; y]$ denotes the concatenation of two column vectors x, y. $\mathbb{S}^n$ denotes the space of $n \times n$ symmetric matrices; and we let $\mathbb{S}^n_+$ be the positive semidefinite cone in $\mathbb{S}^n$. We let I_n denote the identity matrix in $\mathbb{S}^n$. For a matrix $A \in \mathbb{S}^n$, $\lambda_{\max}(A)$, $\|A\|_{\text{Fro}}$, and $\|A\|_{\text{Spec}}$ correspond to its maximal eigenvalue, Frobenius norm, and spectral norm, respectively. Given a set V, we denote its closure by $\operatorname{cl}(V)$. We abuse notation slightly by denoting $\nabla f(x)$ for both the gradient of function f at x if f is differentiable and a subgradient of f at x, even if f is not differentiable. If f is of the form $f(x, u)$, then $\nabla_x f(x, u)$ denotes the subgradient of f at x while keeping the other variables fixed at u.

2.1 Robust Feasibility Problem

Consider a convex *deterministic* or *nominal* mathematical program

$$\min_x \left\{ f^0(x) : \ x \in X, \ \ f^i(x, u^i) \le 0, \ \forall i \in [m] \right\}, \tag{1}$$

where the domain $X \subset \mathbb{R}^n$ is closed and convex, the functions $f^0(x)$ and $f^i(x, u^i)$ for $i \in [m]$ are convex functions of x, and $u = (u^1, \dots, u^m)$ is a fixed parameter vector. Without loss of generality we assume the objective function $f^0(x)$ does not have uncertainty. The *robust convex optimization problem* associated with (1) is

$$\text{Opt} := \min_x \left\{ f^0(x) : \ x \in X, \ \ \sup_{u^i \in U^i} f^i(x, u^i) \le 0, \ \forall i \in [m] \right\}, \tag{2}$$

where $U^1, \dots, U^m$ are the *uncertainty sets* given for the parameter u^i of constraint $i \in [m]$. Because we assume formulation (1) is convex, the overall optimization problem in (2) is convex.

In this paper, we work under the following mild regularity assumption:

Assumption 2.1. *The constraint functions $f^i(x, u^i)$ for all $i \in [m]$ are finite-valued on the domain $X \times U^i$, convex in x and concave in u^i; X, the domain for x, is closed and convex; and U^i, the domains for u^i, are closed and bounded.*

We take Assumption 2.1 as given for all our results and proofs. Without loss of generality, we assume that the uncertainty set has a Cartesian product form $U^1 \times \ldots \times U^m$, see e.g., [8]; we let $U = U^1 \times \ldots \times U^m$ and write $u = [u^1; \ldots; u^m] \in U$. We do not further assume that the sets U^i are convex. However, for some algorithms we consider, convexity of U^i for $i \in [m]$ will be required.

A convex optimization problem can be solved by solving a polynomial number of associated feasibility problems in a standard way, via a binary search over its optimal value. In particular, let $[\underline{\upsilon}_0, \overline{\upsilon}_0]$ be an initial interval containing the optimal value of (2). At each iteration k of the binary search, we update the domain $X_k := X \cap \{x : \ f^0(x) \leq \upsilon_k\}$ for some $\upsilon_k \in [\underline{\upsilon}_k, \overline{\upsilon}_k]$ and arrive at the following robust feasibility problem:

$$\text{find} \quad x \in X_k \quad \text{s.t.} \quad \sup_{u^i \in U^i} f^i(x, u^i) \leq 0 \quad \forall i \in [m]. \tag{3}$$

Then based on the feasibility/infeasibility status of (3), we update our range $[\underline{\upsilon}_{k+1}, \overline{\upsilon}_{k+1}]$ and go to iteration $k+1$. In this scheme, we are guaranteed to find a solution $x^* \in X$ whose objective value is within $\delta > 0$ of the optimum value of (2) in at most $\left\lfloor \log_2 \left(\frac{\overline{\upsilon}_0 - \underline{\upsilon}_0}{\delta} \right) \right\rfloor$ iterations. Therefore, one can equivalently study the complexity of solving robust feasibility problem (3) as opposed to (2). Hence, we focus on solving robust feasibility problem and assume that the constraint on the objective function $f^0(x)$ is already included in the domain X for simplicity in our notation.

Given functional constraints $f^i(x) \leq 0$, $i \in [m]$, most convex optimization methods will declare infeasibility or return an approximate solution $x \in X$ such that $f^i(x) \leq \epsilon$ for $i \in [m]$ for some tolerance level $\epsilon > 0$. Therefore, we consider the following *robust approximate feasibility problem*:

$$\begin{cases} \textit{Either}: \text{ find } \ x \in X \quad \text{s.t.} \quad \sup_{u^i \in U^i} f^i(x, u^i) \leq \epsilon \quad \forall i \in [m]; \\ \textit{or}: \text{ declare infeasibility, } \forall x \in X, \ \exists i \in [m] \quad \text{s.t.} \quad \sup_{u^i \in U^i} f^i(x, u^i) > 0. \end{cases} \tag{4}$$

We refer to any feasible solution x to (4), i.e., $x \in X$ such that $\sup_{u^i \in U^i} f^i(x, u^i) \leq \epsilon$ holds for all $i \in [m]$ as a *robust ϵ-feasibility certificate*. Similarly, any realization of the uncertain parameters $\bar{u} \in U$ such that there exists no $x \in X$ satisfying $f^i(x, \bar{u}^i) \leq 0$ for all $i \in [m]$ is referred to as a *robust infeasibility certificate*.

2.2 Saddle Point Problems

Saddle point (SP) problems play a vital role in our developments. In its most general form, a convex-concave SP problem is given by

$$\text{SV} = \inf_{x \in X} \sup_{y \in Y} \phi(x, y), \tag{$\mathcal{S}$}$$

where the function $\phi(x, y)$ is convex in x and concave in y and the domains X, Y are nonempty closed convex sets in Euclidean spaces $\mathbb{E}_x, \mathbb{E}_y$.

Any convex-concave SP problem ($\mathcal{S}$) gives rise to two convex optimization problems that are dual to each other:

$$\begin{array}{rcll} \text{Opt}(P) & = & \inf_{x \in X} [\overline{\phi}(x) := \sup_{y \in Y} \phi(x, y)] & (P) \\ \text{Opt}(D) & = & \sup_{y \in Y} [\underline{\phi}(y) := \inf_{x \in X} \phi(x, y)] & (D) \end{array}$$

with $\text{Opt}(P) = \text{Opt}(D) = \text{SV}$. It is well-known that the solutions to $(\mathcal{S})$ — the saddle points of ϕ on $X \times Y$ — are exactly the pairs $[x; y]$ formed by optimal solutions to the problems (P) and (D).

We quantify the accuracy of a candidate solution $[\bar{x}, \bar{y}]$ to SP problem $(\mathcal{S})$ with the *saddle point gap* given by

$$\epsilon_{\text{sad}}^{\phi}(\bar{x}, \bar{y}) := \overline{\phi}(\bar{x}) - \underline{\phi}(\bar{y}) = \underbrace{\left[\overline{\phi}(\bar{x}) - \text{Opt}(P)\right]}_{\geq 0} + \underbrace{\left[\text{Opt}(D) - \underline{\phi}(\bar{y})\right]}_{\geq 0}. \tag{5}$$

Because convex-concave SP problems are simply convex optimization problems, they can in principle be solved by polynomial-time interior point methods (IPMs). However, the computational complexity of such methods depends heavily on the dimension of the problem. Thus, scalability of resulting algorithms becomes an issue in large-scale applications. As a result, for large-scale SP problems, one has to resort to first-order subgradient-type methods. On a positive note, there are many efficient first-order methods (FOMs) for convex-concave SP problems. These in particular include Nesterov's accelerated gradient descent algorithm [29] and Nemirovski's Mirror-Prox algorithm [27], both of which bound the saddle point gap at a rate of $\epsilon_{\text{sad}}^{\phi}(\bar{x}_T, \bar{u}_T) \leq O\left(\frac{1}{T}\right)$ where $\bar{x}_T, \bar{u}_T$ are solutions obtained after T iterations.

2.3 Online Convex Optimization Tools

Our efficient approach employs tools from the online convex optimization domain. We now briefly outline; we refer to [15, 17, 32] for further details and applications of OCO.

OCO is used to capture decision making in dynamic environments. We are given a finite time horizon T, closed, bounded, and convex domain Z, and in each time period $t \in [T]$, a convex loss function $f_t : Z \to \mathbb{R}$ is revealed. At time periods $t \in [T]$ we must choose a decision $z_t \in Z$, and based on this we suffer a loss of $f_t(z_t)$ and receive some feedback typically in the form of first-order information on f_t. Our goal is to minimize the *weighted regret*

$$\sum_{t=1}^{T} \theta_t f_t(z_t) - \inf_{z \in Z} \sum_{t=1}^{T} \theta_t f_t(z), \tag{6}$$

where $\theta \in \Delta_T$ is a vector of convex combination weights.[1]

Most OCO algorithms are closely related to offline iterative FOMs. In this paper, we will make use of the proximal setup of [22] to choose the sequence $\{z_t\}_{t=1}^{T}$ which ensures that the weighted regret (6) converges to 0 as $T \to \infty$. Thus, we make the following assumption on Z for the existence of a proximal setup.

Assumption 2.2. *Let $\mathbb{E}_z$ be the Euclidean space containing Z. There exists a norm $\|\cdot\|$ and its dual norm $\|\cdot\|_*$ on $\mathbb{E}_z$, a distance-generating function $\omega : Z \to \mathbb{R}$ which is 1-strongly convex with respect to $\|\cdot\|$ and leads to an easy-to-compute prox function $\text{Prox}_z(\xi) := \arg\min_{w \in Z} \{\langle \xi, w \rangle + \omega(w) - \langle \omega'(z), w - z \rangle\}$ and set width $\Omega := \max_{z \in Z} \omega(z) - \min_{z \in Z} \omega(z)$ which is finite when Z is bounded.*

The proximal setup of Assumption 2.2 allows us to adjust to the geometry of domain Z. The standard basic domains satisfying Assumption 2.2 include simplex, Euclidean ball, and spectahedron; see [22, Section 1.7] for the standard proximal setups (i.e., Assumption 2.2) for these basic domains in terms of selection of $\|\cdot\|$ and resulting ω, Prox computation, and set width Ω.

[1] Note that in the OCO literature, regret is usually defined with uniform weights $\theta_t = 1/T$. Nonuniform weights introduce flexibility to our framework by allowing selection of specific customization of OCO algorithms for exploiting structural properties of the constraint functions f^i to achieve better convergence rates. A prime example for this is when the functions are strongly convex.

Under Assumption 2.2 and various structural properties, the straightforward extension of the standard online mirror descent algorithm (see, e.g., [24]) from uniform weigths to weighted regret achieves the following convergence rate.

Theorem 2.1 ([24, Theorem 5]). *Suppose there exists $G \in (0, \infty)$ such that $\|\nabla f_t(z)\|_* \le G$ for all $z \in Z$, $t \in [T]$. Define $\gamma = \sqrt{\frac{2\Omega}{G^2 \sum_{t=1}^T \theta_t^2}}$. Choose $z_1 = \arg\min_{z \in Z} \omega(z)$ and $z_{t+1} = \mathrm{Prox}_{z_t}(\gamma \theta_t \nabla f_t(z_t))$ for $t \in [T]$. Then*

$$\sum_{t=1}^T \theta_t f_t(z_t) - \inf_{z \in Z} \sum_{t=1}^T \theta_t f_t(z) \le \sqrt{2\Omega G^2 \sum_{t=1}^T \theta_t^2}.$$

In particular, for uniform weights $\theta_t = 1/T$, the upper-bound becomes $O(1/\sqrt{T})$.

We refer to [24] for details of the proof. When $\omega(z) = z^\top z/2$ and weights $\theta_t = 1/T$ for $t \in [T]$, the update rule $z_{t+1} = \mathrm{Prox}_{z_t}(\gamma \nabla f_t(z_t))$ becomes simply gradient descent, and Theorem 2.1 reduces to the standard bound of online gradient descent from [36].

3 General Framework for Robust Feasibility Problems

In this section, we build a general framework to solve the robust feasibility problem (4) by working with its natural saddle point formulation.

Given constraint functions $f^i(x, u^i)$, $i \in [m]$, let us define $\Phi(x, u) := \max_{i \in [m]} f^i(x, u^i)$. Then $\Phi(x, u)$ is a convex function of x, but not necessarily concave in u. In addition, with this definition of $\Phi(\cdot)$, the robust approximate feasibility problem (4) is equivalent to simply verifying

$$\text{either} \quad \inf_{x \in X} \sup_{u \in U} \Phi(x, u) = \inf_{x \in X} \max_{i \in [m]} \sup_{u^i \in U^i} f^i(x, u^i) \le \epsilon \quad \text{or} \quad \inf_{x \in X} \sup_{u \in U} \Phi(x, u) > 0, \tag{7}$$

which is nothing but solving a specific SP problem and checking its value. Analogous to the convex-concave SP gap (5), for a given solution $[\bar{x}, \bar{u}]$, we define the SP gap of problem (7) as

$$\epsilon_{\mathrm{sad}}^{\Phi}(\bar{x}, \bar{u}) := \overline{\Phi}(\bar{x}) - \underline{\Phi}(\bar{u}) = \sup_{u \in U} \Phi(\bar{x}, u) - \inf_{x \in X} \Phi(x, \bar{u}).$$

In general, solving a convex-nonconcave SP problem of form (7), i.e., finding a solution $[\bar{x}, \bar{u}]$ such that $\epsilon_{\mathrm{sad}}^{\Phi}(\bar{x}, \bar{u}) \le \epsilon$, can be difficult. That said, a bound on the SP gap $\epsilon_{\mathrm{sad}}^{\Phi}(\bar{x}, \bar{u})$ along with the value of $\Phi(\bar{x}, \bar{u})$ leads to robust feasibility certificates for (7) as follows.

Theorem 3.1. *Let $\Psi : X \times U \to \mathbb{R}$ be a given function associated with an SP (not necessarily admitting a convex-concave structure). Suppose we have $\bar{x} \in X$, $\bar{u} \in U$, and $\tau \in (0, 1)$ such that $\epsilon_{\mathrm{sad}}^{\Psi}(\bar{x}, \bar{u}) \le \tau\epsilon$. Then if $\Psi(\bar{x}, \bar{u}) \le (1 - \tau)\epsilon$, we have $\sup_{u \in U} \Psi(\bar{x}, u) \le \epsilon$. Moreover, if $\Psi(\bar{x}, \bar{u}) > (1 - \tau)\epsilon$ and $\tau \le \frac{1}{2}$, we have $\inf_{x \in X} \Psi(x, \bar{u}) > 0$.*

Proof. Suppose $\Psi(\bar{x}, \bar{v}) \le (1 - \tau)\epsilon$. Because $\epsilon_{\mathrm{sad}}^{\Psi}(\bar{x}, \bar{v}) = \sup_{u \in U} \Psi(\bar{x}, u) - \inf_{x \in X} \Psi(x, \bar{u}) \le \tau\epsilon$, we have $\sup_{u \in U} \Psi(\bar{x}, u) \le \inf_{x \in X} \Psi(x, \bar{u}) + \tau\epsilon \le \Psi(\bar{x}, \bar{u}) + \tau\epsilon \le \epsilon$. On the other hand, when $\Psi(\bar{x}, \bar{u}) > (1 - \tau)\epsilon$, we have $(1 - \tau)\epsilon < \Psi(\bar{x}, \bar{u}) \le \sup_{u \in U} \Psi(\bar{x}, u) \le \inf_{x \in X} \Psi(x, \bar{u}) + \tau\epsilon$, which implies $\inf_{x \in X} \sup_{u \in U} \Psi(x, u) \ge \inf_{x \in X} \Psi(x, \bar{u}) > (1 - 2\tau)\epsilon \ge 0$ when $\tau \le \frac{1}{2}$. □

Remark 3.1. When $m = 1$, $\Phi(x,u) = f^1(x,u^1)$, and it is thus convex in x and concave in u due to Assumption 2.1. Therefore, in the case of a single robust constraint, i.e., $m = 1$, under Assumption 2.1 and assuming $U = U^1$ is a closed convex set, the optimization problem in (7) reduces to a standard convex-concave SP problem. ∎

While it is not very common, a few robust convex optimization problems come with a single robust constraint and convex uncertainty set U; see for example [2] for a robust version of a SVM problem with one constraint. In such cases, based on Remark 3.1, the resulting convex-concave SP problems can directly be solved via efficient FOMs. On the other hand, in the presence of multiple constraints, the function $\Phi(x,u)$ is not concave in $u = [u^1;\dots;u^m]$ even under Assumption 2.1. Nevertheless, when $m > 1$, it is still possible to have a convex-concave SP reformulation of the optimization problem in (7) in an extended space via perspective transformations, which we present in Appendix A. While this reformulation has the benefit of reducing the robust feasibility problem to a well-known and well-studied problem, it destroys the simplicity of the original domains and constraint functions and hence comes with some challenges. Therefore, we develop a framework where we work directly with the convex-nonconcave SP formulation in (7) in the space of original variables. Moreover, because we work in the original space of variables, we simply utilize the first-order information on the original constraint functions f^i and original domains X and U^i. This direct approach in particular allows us to take greater advantage of the structure of the original formulation such as the availability of efficient projection (prox) computations over domains X, U^i, and/or better parameters for smoothness, Lipschitz continuity, etc., of the functions f^i.

Because $\Phi(x,u)$ is not concave in u, we cannot bound the SP gap $\epsilon^{\Phi}_{\text{sad}}(\bar{x},\bar{u})$ by using traditional FOMs designed for solving convex-concave SP problems. However, we next show that by just *partially* upper bounding $\epsilon^{\Phi}_{\text{sad}}(\bar{x},\bar{u})$, we can derive a general iterative framework to obtain robust feasibility/infeasibility certificates. Further specifics of this framework is described in Section 4.

Henceforth we will no longer use the shorthand notation $\Phi(x,u) = \max_{i\in[m]} f^i(x,u^i)$, but we will denote the SP gap $\epsilon^{\Phi}_{\text{sad}}(\bar{x},\bar{u})$ as

$$\epsilon(\bar{x},\bar{u}) := \epsilon^{\Phi}_{\text{sad}}(\bar{x},\bar{u}) = \max_{i\in[m]} \sup_{u^i\in U^i} f^i(\bar{x},u^i) - \inf_{x\in X} \max_{i\in[m]} f^i(x,\bar{u}^i). \tag{8}$$

The robust feasibility certificate result from Theorem 3.1 indicates the importance of bounding the SP gap $\epsilon(\bar{x},\bar{u})$. Often, FOMs achieve this by iteratively generating points $x_t \in X$, $u_t \in U$ for $t \in [T]$ and tracking the points $\bar{x}$ and $\bar{u}$ obtained from a convex combination of $\{x_t,u_t\}_{t=1}^T$. In order to simplify our notation, given convex combination weights $\theta \in \Delta_T$ and points $\{x_t,u_t\}_{t=1}^T$, we let

$$\bar{x}_T := \sum_{t=1}^T \theta_t x_t \quad \text{and} \quad \bar{u}_T := \sum_{t=1}^T \theta_t u_t.$$

We now present an upper bound on $\epsilon(\bar{x}_T,\bar{u}_T)$ that follows naturally from the convex-concave structure of functions f^i. To this end, given a set of vectors $y_t \in \Delta_m$ for $t \in [T]$, we also define

$$\epsilon^{\circ}(\{x_t,u_t,\theta_t\}_{t=1}^T) := \max_{i\in[m]} \left\{ \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t,u^i) - \sum_{t=1}^T \theta_t f^i(x_t,u_t^i) \right\}, \quad \text{and}$$

$$\epsilon^{\bullet}(\{x_t,u_t,y_t,\theta_t\}_{t=1}^T) := \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t,u_t^i) - \inf_{x\in X} \sum_{t=1}^T \theta_t \sum_{i=1}^m y_t^{(i)} f^i(x,u_t^i),$$

together with

$$\widehat{\epsilon}(\{x_t, u_t, y_t, \theta_t\}_{t=1}^T) := \inf_{x\in X} \sum_{t=1}^T \theta_t \sum_{i=1}^m y_t^{(i)} f^i(x, u_t^i) - \inf_{x\in X} \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x, u_t^i).$$

Our next result relates these quantities to the value of the SP gap function $\epsilon(\bar{x}_T, \bar{u}_T)$.

Proposition 3.1. *Let $x_t \in X$ and $u_t \in U$ for $t \in [T]$ be given a set of vectors. Then for any set of vectors $y_t \in \Delta_m$ for $t \in [T]$ and any $\theta \in \Delta_T$, we have*

$$\epsilon\left(\sum_{t=1}^T \theta_t x_t, \sum_{t=1}^T \theta_t u_t\right) \le \epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T) + \epsilon^\bullet(\{x_t, u_t, y_t, \theta_t\}_{t=1}^T) + \widehat{\epsilon}(\{x_t, u_t, y_t, \theta_t\}_{t=1}^T). \quad (9)$$

Proof. Given $y_t \in \Delta_m$ for $t \in [T]$ and $\theta \in \Delta_T$, let us define $\bar{x} := \sum_{t=1}^T \theta_t x_t$ and $\bar{u} := \sum_{t=1}^T \theta_t u_t$. We first partition $\epsilon(\bar{x}, \bar{u})$ as $\epsilon(\bar{x}, \bar{u}) = \overline{\epsilon}(\bar{x}, \bar{u}) + \underline{\epsilon}(\bar{x}, \bar{u})$ where

$$\begin{aligned}
\overline{\epsilon}(\bar{x}, \bar{u}) &:= \max_{i\in[m]} \sup_{u^i\in U^i} f^i(\bar{x}, u^i) - \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i), \\
\underline{\epsilon}(\bar{x}, \bar{u}) &:= \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) - \inf_{x\in X} \max_{i\in[m]} f^i(x, \bar{u}^i),
\end{aligned}$$

and then derive upper bounds on $\overline{\epsilon}(\bar{x}, \bar{u})$ and $\underline{\epsilon}(\bar{x}, \bar{u})$.

We start with bounding $\overline{\epsilon}(\bar{x}, \bar{u})$. Because the functions $f^i(x, u^i)$ are convex in x for all i and $\theta \in \Delta_T$, we have $\max_{i\in[m]} \sup_{u^i\in U^i} f^i(\bar{x}, u^i) \le \max_{i\in[m]} \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i)$. Therefore,

$$\begin{aligned}
\overline{\epsilon}(\bar{x}, \bar{u}) &= \max_{i\in[m]} \sup_{u^i\in U^i} f^i(\bar{x}, u^i) - \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) \\
&\le \max_{i\in[m]} \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) + \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) - \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) \\
&\le \max_{i\in[m]} \left\{ \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \right\} + \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) - \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i),
\end{aligned} \quad (10)$$

where the last inequality follows since $\max_{i\in[m]}\{\alpha_i - \beta_i\} \ge \max_{i\in[m]} \alpha_i - \max_{i\in[m]} \beta_i$ for any sequence of numbers α_i, β_i, $i \in [m]$.

Note that $\inf_{x\in X} \max_{i\in[m]} f^i(x, \bar{u}^i) \ge \inf_{x\in X} \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x, u^i)$ because under Assumption 2.1 the functions $f^i(x, u^i)$ are concave in u^i for all i. Thus, we arrive at

$$\begin{aligned}
\underline{\epsilon}(\bar{x}, \bar{u}) &= \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) - \inf_{x\in X} \max_{i\in[m]} f^i(x, \bar{u}^i) \\
&\le \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) - \inf_{x\in X} \sum_{t=1}^T \theta_t \sum_{i=1}^m y_t^{(i)} f^i(x, u_t^i) + \inf_{x\in X} \sum_{t=1}^T \theta_t \sum_{i=1}^m y_t^{(i)} f^i(x, u_t^i) \\
&\qquad - \inf_{x\in X} \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x, u_t^i) \\
&= \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) - \inf_{x\in X} \sum_{t=1}^T \theta_t \sum_{i=1}^m y_t^{(i)} f^i(x, u_t^i) + \widehat{\epsilon}(\{x_t, u_t, y_t, \theta_t\}_{t=1}^T).
\end{aligned} \quad (11)$$

Then by summing (10) and (11) and rearranging the terms, we deduce the result. □

We are now ready to state our main result. This is analogous to Theorem 3.1 except that we do not need to bound all three terms in (9), but instead it suffices to guarantee that

$$\epsilon^{\circ}(\{x_t, u_t, \theta_t\}_{t=1}^T) + \epsilon^{\bullet}(\{x_t, u_t, y_t, \theta_t\}_{t=1}^T) \le \epsilon.$$

When this holds, based on the value of $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i)$ we can then obtain robust ϵ-feasibility/infeasibility certificates.

Theorem 3.2. *Suppose we have sequences $\{x_t, u_t, y_t, \theta_t\}_{t=1}^T$ with $x_t \in X$, $u_t \in U$, $y_t \in \Delta_m$ for all $t \in [T]$, $\theta \in \Delta_T$. Let $\tau \in (0, 1)$. If $\epsilon^{\circ}(\{x_t, u_t, \theta_t\}_{t=1}^T) \le \tau\epsilon$ and $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \le (1-\tau)\epsilon$, then the solution $\bar{x}_T := \sum_{t=1}^T \theta_t x_t$ is ϵ-feasible with respect to (4). If $\epsilon^{\bullet}(\{x_t, u_t, y_t, \theta_t\}_{t=1}^T) \le (1-\tau)\epsilon$ and $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) > (1-\tau)\epsilon$, then (4) is infeasible.*

Proof. First suppose there exists a $\tau \in (0, 1)$ and corresponding vectors $\{x_t, u_t, y_t, \theta_t\}_{t=1}^T$ such that $\epsilon^{\circ}(\{x_t, u_t, \theta_t\}_{t=1}^T) \le \tau\epsilon$ and $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \le (1-\tau)\epsilon$ holds as well. Note that

$$\begin{aligned}\tau\epsilon \ge \epsilon^{\circ}(\{x_t, u_t, \theta_t\}_{t=1}^T) &= \max_{i\in[m]} \left\{ \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \right\} \\ &\ge \max_{i\in[m]} \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i), \qquad (12)\end{aligned}$$

where the last inequality follows since $\max_{i\in[m]}\{\alpha_i - \beta_i\} \ge \max_{i\in[m]} \alpha_i - \max_{i\in[m]} \beta_i$ for any sequence of numbers α_i, β_i, $i \in [m]$. Then $\bar{x}_T$ is an ϵ-feasible solution for (4) because

$$\begin{aligned}\max_{i\in[m]} \sup_{u^i\in U^i} f^i\left(\bar{x}_T, u^i\right) &= \max_{i\in[m]} \sup_{u^i\in U^i} f^i\left(\sum_{t=1}^T \theta_t x_t, u^i\right) \\ &\le \max_{i\in[m]} \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) \le \tau\epsilon + \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \le \epsilon,\end{aligned}$$

where the first inequality follows from the convexity of the functions f^i and the fact that $\theta \in \Delta_T$, the second inequality from (12), and the last inequality holds since $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \le (1-\tau)\epsilon$.

On the other hand, suppose $\epsilon^{\bullet}(\{x_t, u_t, y_t, \theta_t\}_{t=1}^T) \le (1-\tau)\epsilon$ and $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) > (1-\tau)\epsilon$. Note that

$$\begin{aligned}\inf_{x\in X} \sum_{t=1}^T \theta_t \sum_{i=1}^m y_t^{(i)} f^i(x, u_t^i) &\le \inf_{x\in X} \sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x, u_t^i) \\ &\le \inf_{x\in X} \sum_{t=1}^T \theta_t \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) = \inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i), \qquad (13)\end{aligned}$$

where the first inequality follows since $y_t \in \Delta_m$ for all $t \in [T]$, the second inequality holds because $f^i(x, u_t^i) \le \sup_{u^i\in U^i} f^i(x, u^i)$ for all $i \in [m]$ and $y_t^{(i)} \ge 0$ for $i \in [m]$, $t \in [T]$, and the last equation follows from $\theta \in \Delta_T$. Then using the bound

$$(1-\tau)\epsilon \ge \epsilon^{\bullet}(\{x_t, u_t, y_t, \theta_t\}_{t=1}^T) = \max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) - \inf_{x\in X} \sum_{t=1}^T \theta_t \sum_{i=1}^m y_t^{(i)} f^i(x, u_t^i), \qquad (14)$$

we arrive at

$$\inf_{x\in X}\max_{i\in[m]}\sup_{u^i\in U^i} f^i(x,u^i) \geq \inf_{x\in X}\sum_{t=1}^T \theta_t \sum_{i=1}^m y_t^{(i)} f^i(x,u_t^i) \geq \max_{i\in[m]}\sum_{t=1}^T \theta_t f^i(x_t,u_t^i) - (1-\tau)\epsilon > 0,$$

where the first inequality follows from inequality (13), the second inequality from (14) and the last inequality holds because $\max_{i\in[m]}\sum_{t=1}^T \theta_t f^i(x_t,u_t^i) > (1-\tau)\epsilon$. This implies (4) is infeasible. □

In Section 4.1 we will show that $\epsilon^\circ(\{x_t,u_t,\theta_t\}_{t=1}^T)$ can be interpreted as a weighted regret term (6). On the other hand, the term $\epsilon^\bullet(\{x_t,u_t,y_t,\theta_t\}_{t=1}^T)$ has no such direct interpretation. In order to upper-bound it by a weighted regret term, we need the following result.

Corollary 3.1. *Given sequences $\{x_t,u_t,\theta_t\}_{t=1}^T$ with $x_t\in X$, $u_t\in U$, for all $t\in[T]$, $\theta\in\Delta_T$, there is an appropriate choice of sequence $\{\bar{y}_t\}_{t=1}^T$ where $\bar{y}_t\in\Delta_m$ for all $t\in[T]$, such that $\epsilon^\bullet(\{x_t,u_t,\bar{y}_t,\theta_t\}_{t=1}^T)$ is upper-bounded by*

$$\epsilon^\bullet(\{x_t,u_t,\theta_t\}_{t=1}^T) := \sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x_t,u_t) - \inf_{x\in X}\sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x,u_t). \tag{15}$$

Thus, if $\epsilon^\bullet(\{x_t,u_t,\theta_t\}_{t=1}^T) \leq (1-\tau)\epsilon$ and $\max_{i\in[m]}\sum_{t=1}^T \theta_t f^i(x_t,u_t^i) > (1-\tau)\epsilon$, then (4) is infeasible.

Proof. Given $\{u_t\}_{t=1}^T$, let $x^*\in\arg\min_{x\in X}\sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x,u_t^i)$ and for all $t\in[T]$ define $\bar{y}_t\in\mathbb{R}^m$ to be the i-th unit vector where i is the smallest index satisfying $i\in\arg\max_{i'\in[m]} f^i(x^*,u_t^{i'})$. Then $\inf_{x\in X}\sum_{t=1}^T \theta_t \sum_{i=1}^m \bar{y}_t^{(i)} f^i(x,u_t^i) = \inf_{x\in X}\sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x,u_t^i)$, and the bound follows from $\max_{i\in[m]}\sum_{t=1}^T \theta_t f^i(x_t,u_t) \leq \sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x_t,u_t)$. The last result follows from Theorem 3.2. □

The following corollary demonstrates how we can choose τ.

Corollary 3.2. *Suppose $\{x_t,u_t,y_t,\theta_t\}_{t=1}^T$ with $x_t\in X$, $u_t\in U$, $y_t\in\Delta_m$ for all $t\in[T]$, and $\theta\in\Delta_T$ is such that there exists $\kappa^\circ,\kappa^\bullet\in(0,1)$ satisfying $\epsilon^\circ(\{x_t,u_t,\theta_t\}_{t=1}^T)\leq\epsilon\,\kappa^\circ$ and $\epsilon^\bullet(\{x_t,u_t,y_t,\theta_t\}_{t=1}^T)\leq\epsilon\,\kappa^\bullet$ with $\kappa^\circ+\kappa^\bullet\leq 1$. Let $\tau\in[\kappa^\circ,1-\kappa^\bullet]$. Whenever $\max_{i\in[m]}\sum_{t=1}^T \theta_t f^i(x_t,u_t^i)\leq(1-\tau)\epsilon$ as well, the solution $\bar{x}_T := \sum_{t=1}^T \theta_t x_t$ is ϵ-feasible with respect to (4). Also, whenever $\max_{i\in[m]}\sum_{t=1}^T \theta_t f^i(x_t,u_t^i) > (1-\tau)\epsilon$, then (4) is infeasible.*

Proof. Note that $\tau\in(0,1)$ follows from its definition, $\kappa^\circ,\kappa^\bullet\geq 0$, and $\kappa^\circ+\kappa^\bullet\leq 1$. Furthermore, the interval $[\kappa^\circ,1-\kappa^\bullet]$ is well-defined since $\kappa^\circ\leq 1-\kappa^\bullet$ always holds. Moreover, $\epsilon^\circ(\{x_t,u_t,\theta_t\}_{t=1}^T)\leq\epsilon\,\kappa^\circ\leq\epsilon\,\tau$ and $\epsilon^\bullet(\{x_t,u_t,y_t,\theta_t\}_{t=1}^T)\leq\epsilon\,\kappa^\bullet\leq\epsilon(1-\tau)$ holds from the definition of τ. The result now follows from Theorem 3.2. □

Theorem 3.2 and Corollary 3.1 points to our general iterative framework for finding robust feasibility/infeasibility certificates of (4): generate sequences $\{x_t,u_t\}_{t=1}^T$ iteratively to bound $\epsilon^\circ(\{x_t,u_t,\theta_t\}_{t=1}^T)$ and $\epsilon^\bullet(\{x_t,u_t,\theta_t\}_{t=1}^T)$, and then evaluate the term $\max_{i\in[m]}\sum_{t=1}^T \theta_t f^i(x_t,u_t^i)$. In the next section, we describe some approaches to generate these sequences in practice.

4 Customizations of the General Framework

In this section, we examine how to generate the sequences $\{x_t, u_t\}_{t=1}^T$ in practice. In Section 4.1, we first interpret the terms $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T)$ and $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$ from Section 3 as weighted regret terms, which gives rise to our OFO-based approach. In Section 4.2, we modify the pessimization oracle-based approach of [26] to solving (4) within our framework. In Section 4.3, we examine the nominal feasibility oracle-based approach of [5] within the context of our general framework. Finally, in Section 4.4, we summarize and compare the convergence rates achievable via various customizations of these different approaches.

4.1 The OFO-based Approach

Let us first consider $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T)$. For any $i \in [m]$, given x_t, we define the function $f_t^i : U^i \to \mathbb{R}$ as $f_t^i(u^i) = -f^i(x_t, u^i)$. Then the function $f_t^i(u^i)$ is convex in u^i under Assumption 2.1, and the subterm of $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T)$ given by

$$\sup_{u^i \in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \tag{16}$$

is the weighted regret (6) corresponding to the sequence of functions $\{f_t^i\}_{t=1}^T$. When the uncertainty sets U^i, $i \in [m]$ admit proximal setups as in Assumption 2.2, Theorem 2.1 from Section 2.3 gives an efficient OFO algorithm for choosing $\{u_t^i\}_{i=1}^m$ to bound the regret subterms (16) with $O(1/\sqrt{T})$. Therefore, by choosing T sufficiently large, i.e., $T = O(1/\epsilon^2)$, and employing one of these algorithms for each constraint $i \in [m]$, we can generate a sequence $\{u_t\}_{t=1}^T$ that guarantees $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T) \leq \tau\epsilon$ for *any* sequence $\{x_t\}_{t=1}^T$.

On the other hand, given $u_t^i \in U^i$ for $i \in [m]$, let us define $\varphi_t(x) := \max_{i\in[m]} f^i(x, u_t^i)$. Then $\varphi_t(x)$ is convex in x over X since the functions f^i are convex in x by Assumption 2.1. We can then rewrite $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$ as

$$\begin{aligned}\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T) &= \sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x_t, u_t^i) - \inf_{x\in X} \sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x, u_t^i) \\ &= \sum_{t=1}^T \theta_t \varphi_t(x_t) - \inf_{x\in X} \sum_{t=1}^T \theta_t \varphi_t(x). \end{aligned} \tag{17}$$

Then $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$ is also a weighted regret term (6) corresponding to the sequence of functions $\{\varphi_t\}_{t=1}^T$. When the domain X admits a proximal setup as in Assumption 2.2, Theorem 2.1 again gives an efficient OFO algorithm for choosing x_t to bound (17). Thus, by choosing T sufficiently large and employing one of these algorithms, we can generate a sequence $\{x_t\}_{t=1}^T$ to guarantee $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T) \leq (1-\tau)\epsilon$ for *any* sequence $\{u_t\}_{t=1}^T$.

Our OFO-based approach can thus be described as follows: choose T sufficiently large, then *simultaneously* generate $\{u_t\}_{t=1}^T$ and $\{x_t\}_{t=1}^T$ iteratively using tools from Section 2.3 to bound $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T) \leq \tau\epsilon$ and $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T) \leq (1-\tau)\epsilon$. Thereby, via Theorem 3.2 and Corollary 3.1, we can solve the robust feasibility problem (4) using only cheap OFO algorithms, which avoids relying on a pessimization oracle for u as in [26] or a nominal feasibility oracle for x as in [5].

Remark 4.1. Since we are generating both x_t and u_t in an online manner, a conflict may arise if we encounter a situation where generating x_t requires knowledge of u_t, and generating u_t also requires knowledge of x_t. However, observe that Theorem 2.1 generates in a *non-anticipatory manner* the current decision z_t only with knowledge of f_{t-1}, and not of f_t. In the context of our general framework, we will only use u_{t-1} to generate x_t, and similarly we only use x_{t-1} to generate u_t, thus no conflicts will arise. ∎

Our OFO-based approach is quite flexible in terms of the selection of OFO algorithms, and is certainly not restricted to only using online mirror descent as in Theorem 2.1. To accommodate this flexibility, we describe our OFO-based approach for obtaining robust feasibility/infeasibility certificates for (4) precisely in Algorithm 1 for general choices of OFO algorithms. For this, we let $\mathcal{A}_i$ be OFO algorithms that work with domains U^i for each $i \in [m]$ and bound the regret terms (16), and $\mathcal{A}_x$ be an OFO algorithm that works with the domain X to bound the regret term (17).

Algorithm 1 OFO-based approximate robust feasibility solver.

input: tolerance level $\epsilon > 0$, sufficiently large $T = T(\epsilon)$ and convex combination weights $\theta_1, \dots, \theta_T > 0$ to achieve $\epsilon/2$ weighted regret guarantees from $\mathcal{A}_i$ and $\mathcal{A}_x$.
output: either $\bar{x} \in X$ such that $\sup_{u^i \in U^i} f^i(\bar{x}, u^i) \le \epsilon$ for all $i \in [m]$, or an infeasibility certificate for (4).
initialize $u_0^i \in U^i$ for $i \in [m]$ and $x_0 \in X$ according to $\mathcal{A}_i, \mathcal{A}_x$ respectively.
for $t = 1, \dots, T$ **do**
  for $i = 1, \dots, m$ **do**
    update $u_t^i \in U^i$ according to $\mathcal{A}_i$ using loss functions $f_s^i(u) = -f^i(x_s, u)$, $s \in [t-1]$.
  end for
  update $x_t \in X$ according to $\mathcal{A}_x$ using loss functions $\varphi_s(x) = \max_{i\in[m]} f^i(x, u_s^i)$, $s \in [t-1]$.
  obtain upper bounds $\epsilon\kappa_t^\circ \ge \epsilon^\circ(\{x_s, u_s, \theta_s\}_{s=1}^t)$, $\epsilon\kappa_t^\bullet \ge \epsilon^\bullet(\{x_s, u_s, \theta_s\}_{s=1}^t)$ from $\mathcal{A}_i$, $\mathcal{A}_x$ respectively.
  if $\kappa_t^\circ + \kappa_t^\bullet \le 1$ **then**
    set $\vartheta_t := \max_{i\in[m]} \sum_{s=1}^t \theta_s f^i(x_s, u_s^i)$ and $\tau_t := 1 - \kappa_t^\bullet$.
    if $\vartheta_t > (1-\tau_t)\epsilon$ **then return** 'infeasible'.
    if $\vartheta_t \le (1-\tau_t)\epsilon$ **then return** $\bar{x}_t = \frac{1}{t}\sum_{s=1}^t x_s$ as a robust ϵ-feasible solution to (4).
  end if
end for

Remark 4.2. Note that Algorithm 1 chooses $\tau_t = 1 - \kappa_t^\bullet$, whereas Corollary 3.2 allows us to choose from a range $\tau_t \in [\kappa_t^\circ, 1-\kappa_t^\bullet]$. This is because it is theoretically possible for (4) to simultaneously be infeasible and robust ϵ-feasible, but in practice we would like to discover infeasibility of (4) rather than an approximately feasible solution. Then the best value for $\tau_t \in [\kappa_t^\circ, 1-\kappa_t^\bullet]$ in detecting infeasibility of (4) is given by $\tau_t = 1 - \kappa_t^\bullet$. ∎

4.2 The Pessimization Oracle-Based Approach

Mutapcic and Boyd [26] generate solutions $x_t \in X$ at each iteration t by solving an extended nominal problem

$$\min_{x\in X} \left\{ f^0(x) : \ f^i(x, u^i) \le 0, \ \forall u^i \in \hat{U}_{t-1}^i, \ i \in [m] \right\}, \tag{18}$$

where $\hat{U}_{t-1}^i \subset U^i$ are finite approximate uncertainty sets based on past noise realizations $\{u_{t'}^i\}_{i=1}^m$ for $t' \in [t-1]$. New noises $\{u_t^i\}_{i=1}^m$ are then generated by calling the *pessimization oracles* on the current solution x_t. More precisely, given $x_t \in X$, the pessimization oracles solve $\sup_{u^i\in U^i} f^i(x_t, u^i)$ and return

$$u_t^i \in U^i \quad \text{s.t.} \quad f^i(x_t, u_t^i) \ge \sup_{u^i\in U^i} f^i(x_t, u^i) - \tau\epsilon. \tag{19}$$

If for all $i \in [m]$ we have $f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$, then we terminate and declare x_t is a robust ϵ-feasible and optimal solution; otherwise, we append $\hat{U}_t^i = \hat{U}_{t-1}^i \cup \{u_t^i\}$ and re-solve (18) with the new approximate sets $\hat{U}_t^i$. It is shown in [26, Section 5.2] that the number of iterations T needed before termination with a robust ϵ-feasible solution x_T is upper bounded by $(1+O(1/\epsilon))^n$ where n is the dimension of x.

Suppose now that we are interested in robust feasibility (4). [26, Section 5.3] discusses a number of variations for generating x_t by modifying (18). In contrast, we propose the following modification: instead of solving (18), generate $\{x_t\}_{t=1}^T$ in a non-anticipatory manner (see Remark 4.1) to bound $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$. Then the pessimization oracle-based approach fits within our framework as a special case.

Theorem 4.1. *Let $\tau \in (0,1)$. Suppose $\{x_t\}_{t=1}^T$ are generated iteratively to guarantee that $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T) \leq (1-\tau)\epsilon$ for any sequence $\{u_t\}_{t=1}^T$. Suppose u_t^i are generated by pessimization oracles (19) for $i \in [m]$. If there exists $t \in [T]$ such that for all $i \in [m]$ we have $f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$, then x_t is a robust ϵ-feasible solution to (4). If $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$, then $\bar{x}_T = \sum_{t=1}^T \theta_t x_t$ is a robust ϵ-feasible solution to (4). If $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) > (1-\tau)\epsilon$, then we certify that (4) is robust infeasible.*

Proof. It is clear that if there exists $t \in [T]$ such that for all $i \in [m]$ we have $f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$, then x_t is a robust ϵ-feasible solution to (4). Furthermore, the fact that $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) > (1-\tau)\epsilon$ implies robust infeasibility of (4) follows from our assumption that $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T) \leq (1-\tau)\epsilon$ and Corollary 3.1. To show that $\max_{i\in[m]} \sum_{t=1}^T f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$ implies that $\bar{x}_T$ is robust ϵ-feasible, we only need to show that $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T) \leq \tau\epsilon$. Observe that by our definition of u_t^i in (19), we have $f^i(x_t, u_t^i) \geq \sup_{u^i \in U^i} f^i(x_t, u^i) - \tau\epsilon$, hence the regret terms in $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T)$ satisfy

$$\begin{aligned}
\sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) &\leq \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \sum_{t=1}^T \theta_t \left(\sup_{u^i\in U^i} f^i(x_t, u^i) - \tau\epsilon \right) \\
&= \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \sum_{t=1}^T \theta_t \sup_{u^i\in U^i} f^i(x_t, u^i) + \tau\epsilon \\
&\leq \tau\epsilon.
\end{aligned}$$

Then

$$\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T) = \max_{i\in[m]} \left\{ \sup_{u^i\in U^i} \sum_{t=1}^T \theta_t f^i(x_t, u^i) - \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \right\} \leq \tau\epsilon,$$

and the result follows from Corollary 3.1. □

Theorem 4.1 can only be used to certify robust feasibility/infeasibility. Hence, to find a robust ϵ-optimal solution, we must perform a binary search and solve at most $O(\log(1/\epsilon))$ instances of robust feasibility problems. Despite this, in Section 4.4, we discuss how using OFO algorithms to generate $\{x_t\}_{t=1}^T$ to bound $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$ results in much better complexity guarantees than using (18) as proposed by [26], even when taking into account the additional $O(\log(1/\epsilon))$ factor.

Remark 4.3. In the pessimization oracle-based approach, the noises u_t need to be generated with knowledge of x_t, because it is not possible to guarantee $f^i(x_t, u_t^i) \geq \sup_{u^i\in U^i} f^i(x_t, u^i) - \tau\epsilon$ if the vectors u_t^i were chosen with only the knowledge of $x_1, \ldots, x_{t-1}$. ∎

4.3 The Nominal Feasibility Oracle-Based Approach

The nominal feasibility oracle-based approach of Ben-Tal et al. [5] suggest using OFO algorithms to choose a sequence $\{u_t\}_{t=1}^T$ that guarantees $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T)$ is small, in a non-anticipatory fashion, for *any* sequence $\{x_t\}_{t=1}^T$. In this aspect, it essentially matches with our OFO-based approach outlined in Section 4.1. The key differentiating point between our OFO-based approach and that of [5] lies in how the sequence $\{x_t\}_{t=1}^T$ is chosen. At step t, [5] utilizes a *nominal feasibility oracle*. That is, given parameters u_t, they call a powerful, and potentially expensive, nominal feasibility oracle that solves the following feasibility problem to ϵ-accuracy

$$\begin{cases} \textit{Either}\text{: find } \ x \in X \quad \text{s.t.} \quad f^i(x, u_t^i) \le (1-\tau)\epsilon \quad \forall i \in [m]; \\ \textit{or}\text{: declare infeasibility, } \forall x \in X,\ \exists i \in [m] \quad \text{s.t.} \quad f^i(x, u_t^i) > 0. \end{cases} \tag{20}$$

We denote $x_t \in X$ to be the point returned by this oracle at step t, if it exists. For this approach, the outputs of a nominal feasibility oracle can be used to deduce a result similar to Corollary 3.1, except that we no longer need to evaluate $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$, we just need to bound $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T)$.

Theorem 4.2. *Given weights $\theta \in \Delta_T$, suppose that the sequence $\{u_t\}_{t=1}^T$ is generated in a non-anticipatory manner to guarantee $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T) \le \tau\epsilon$ for any sequence $\{x_t\}_{t=1}^T$. Also, suppose that at each step $t \in [T]$, x_t is generated by the nominal feasibility oracle which solves (20). If there exists $t \in [T]$ such that (20) declares infeasibility, then (4) is infeasible. Otherwise, if x_t satisfies $f^i(x_t, u_t^i) \le (1-\tau)\epsilon$ for all $t \in [T]$ and $i \in [m]$, we have a robust ϵ-feasibility certificate for (4).*

Proof. If (20) declares infeasibility, then it is obvious that the robust feasibility problem is infeasible. We focus on the latter case. By the premise of the theorem, we have $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T) \le \tau\epsilon$. Let us evaluate $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i)$. Because $\theta \in \Delta_T$ and from the definition of the nominal feasibility oracle we have $f^i(x_t, u_t^i) \le (1-\tau)\epsilon$ for all $t \in [T]$ and $i \in [m]$, we conclude $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \le (1-\tau)\epsilon$. The conclusion now follows from Theorem 3.2. □

Thus, the approach of [5], which works with nominal feasibility oracles, fits within our framework right away. We next make three important remarks.

Remark 4.4. Similar to Remark 4.3, a critical property required in the approach of [5] of the vectors x_t is that $f^i(x_t, u_t^i) \le (1-\tau)\epsilon$. This is possible only if x_t were chosen with the knowledge of $\{u_1^i\}_{i=1}^m, \ldots, \{u_t^i\}_{i=1}^m$. ∎

Remark 4.5. Theorem 4.2 states that the nominal feasibility oracle-based approach can solve robust feasibility problems (4). This then recovers [5, Theorems 1,2]. In addition, we next make a nice and practical observation that was overlooked in [5]. We show that slightly adjusting this oracle will let us directly solve the robust *optimization* problem (2), i.e., optimize a convex objective function $f^0(x)$ instead of relying on a binary search over the optimal objective value. Recall that Opt is the optimal value of the RO problem (see (2)). Naively, to solve for Opt, we would embed f^0 into the constraint set, and then perform a binary search over the robust feasible set by repeatedly applying the oracle-based approach and Theorem 4.2 to check for robust feasibility. Suppose that now, instead of using a nominal feasibility oracle to solve (20), we work with a *nominal optimization oracle*. That is, given fixed parameters $\{u_t^i\}_{i=1}^m$, we have access to an oracle that solves

$$\text{Opt}_t = \inf_x \left\{ f^0(x) : f^i(x, u_t^i) \le 0,\ i \in [m], \quad x \in X \right\}.$$

When solving for Opt_t, most convex optimization solvers will either declare that the constraints are infeasible, or return a point $x_t \in X$ such that $f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$ and $f^0(x_t) \leq \mathrm{Opt}_t + \epsilon$. It is clear that $f^0(x_t) \leq \mathrm{Opt}_t + \epsilon \leq \mathrm{Opt} + \epsilon$. Given such a sequence of points $\{x_t\}_{t=1}^T$, from Theorem 4.2 we deduced that $\bar{x}_T = \sum_{t=1}^T \theta_t x_t$ is a robust ϵ-feasible solution. Moreover, convexity of f^0 implies

$$f^0(\bar{x}_T) \leq \sum_{t=1}^T \theta_t f^0(x_t) \leq \sum_{t=1}^T \theta_t (\mathrm{Opt} + \epsilon) = \mathrm{Opt} + \epsilon.$$

Hence, not only do we claim that $\bar{x}_T$ is robust ϵ-feasible, but that it is also ϵ-optimal. Thus, when our oracle can return ϵ-optimal solutions, which most solvers can, we eliminate the need to perform a binary search. ∎

Below we elaborate on the differences between Theorem 4.2 and Corollary 3.1.

Remark 4.6. In contrast to Corollary 3.1, Theorem 4.2 does not need to control the term $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$. There are two reasons for this: (i) due to (20), each point x_t satisfies $f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$, hence $\max_{i\in[m]} \sum_{t=1}^T \theta_t f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$ always holds; therefore, the infeasibility part of Corollary 3.1 never becomes relevant; and (ii) due to the oracle solving (20), infeasibility may be declared at any step $t \in [T]$ in Theorem 4.2. This offers the possibility of stopping early rather than having to wait until all T steps are completed. Thus, the nominal feasibility oracle-based approach trades off using more effort at each iteration t to solve (20) for the ability to terminate early. In contrast, our OFO-based approach opts to keep the per-iteration cost cheap while giving up the ability to terminate early. More formally, let us examine a particular way of solving (20) within a nominal feasibility oracle. Note that (20) is equivalent to checking $F_t \leq (1-\tau)\epsilon$ or $F_t > 0$, where

$$F_t := \inf_{x\in X} \left\{ \max_{i\in[m]} f^i(x, u_t^i) \right\}. \tag{21}$$

Since each $f^i(x, u_t^i)$ is convex in x for fixed u_t^i, $\max_{i\in[m]} f^i(x, u_t^i)$ is convex in x also, hence standard convex optimization methods may be employed to find $x_t \in X$ such that

$$F_t \leq \max_{i\in[m]} f^i(x_t, u_t^i) \leq F_t + (1-\tau)\epsilon.$$

Then, by checking whether $\max_{i\in[m]} f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$ or $\max_{i\in[m]} f^i(x_t, u_t^i) > (1-\tau)\epsilon$, we can determine whether $F_t \leq (1-\tau)\epsilon$ or $F_t > 0$ respectively. In particular, if we find that $F_t \leq (1-\tau)\epsilon$, our point x_t is feasible for (20).

Also, when all the vectors x_t satisfy (21), we have the bound

$$\begin{aligned}\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T) &= \sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x_t, u_t^i) - \inf_{x\in X} \sum_{t=1}^T \theta_t \max_{i\in[m]} f^i(x, u_t^i) \\ &\leq \sum_{t=1}^T \theta_t \left[\max_{i\in[m]} f^i(x_t, u_t^i) - \inf_{x\in X} \max_{i\in[m]} f^i(x, u_t^i) \right] \leq (1-\tau)\epsilon.\end{aligned}$$

Consequently, we deduce that the nominal feasibility oracle also naturally bounds $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$ although this bound is not utilized in Theorem 4.2. Note that the term $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$ inherently includes the objective functions $\max_{i\in[m]} f^i(x, u_t^i)$ of each problem F_t. At each iteration t, instead of evaluating F_t to $(1-\tau)\epsilon$ accuracy, our OFO-based approach performs only a simple update based on the first-order information, and it yields a bound on $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$ from the overall collection of these simple updates. ∎

4.4 Convergence Rates and Discussion

We start by summarizing the convergence rates achievable from our OFO-based approach in various cases. We use the notation $r_u(\epsilon)$ to denote the number of iterations required for algorithms $\mathcal{A}_i$ to generate a sequence $\{u_t\}_{t=1}^T$ such that $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T) \le \epsilon/2$. Similarly, we let $r_x(\epsilon)$ be the number of iterations required for $\mathcal{A}_x$ to generate a sequence $\{x_t\}_{t=1}^T$ so that $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T) \le \epsilon/2$. Then the resulting worst-case number of iterations needed in Algorithm 1 to obtain robust ϵ-feasibility/infeasibility certificates is $\max\{r_u(\epsilon), r_x(\epsilon)\}$. In order to discuss the total *arithmetic complexity* of each approach, we also let k be the maximum dimension of the uncertain parameters u^i for $i \in [m]$ and n denotes the dimension of the decision variables x.

As outlined in Section 4, employing standard OFO-based algorithms, i.e., Theorem 2.1, on the terms $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t=1}^T)$ and $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T)$ requires $r_u(\epsilon) = O(1/\epsilon^2)$ and $r_x(\epsilon) = O(1/\epsilon^2)$ iterations to ensure they are no larger than $\epsilon/2$. Furthermore, OFO-based approach returns only robust ϵ-feasible solutions, thus to obtain ϵ-optimal solutions, we need to perform a binary search and repeatedly invoke our method $O(\log(1/\epsilon))$ times. Furthermore, in the case where our domains $X, \{U^i\}_{i=1}^m$ have favorable geometry, such as Euclidean ball or simplex, the vectors $x_t, y_t, \{u_t^i\}_{i=1}^m$ are updated via simple closed-form prox operations, which cost $O(km+mn)$ per iteration. Because the cost of computing the subgradients $\nabla_x f^i(x,u^i), \nabla_u f^i(x,u^i)$ is incurred in each iteration of all of the algorithms, we disregard these in our comparison.

Remark 4.7. The flexibility of our approach extends beyond just using Theorem 2.1. Depending on the structure of functions f^i and uncertainty domains U^i, the algorithms $\mathcal{A}_i$ and $\mathcal{A}_x$ may be replaced by more appropriate OCO algorithms. For example, when f^i are strongly convex, certain OCO algorithms achieve faster convergence rates. Moreover, unless explicitly required by the algorithms $\mathcal{A}_i$, we do not need to assume convexity of the sets U^i. As a result, the follow-the-leader or follow-the-perturbed-leader type algorithms from [25] can be utilized as $\mathcal{A}_i$ in our framework when U^i are nonconvex under certain assumptions ensuring applicability of these algorithms. Such assumptions are satisfied for example when $f^i(x,u^i)$ are linear in u^i and the nonconvex sets U^i admit a certain linear optimization oracle. Similarly, when the functions $f^i(x,u^i)$ are exp-concave in u^i, applying the online Newton step algorithm of [18] for $\mathcal{A}_i$ results in a weighted regret bound of at most $O(\log(T)/T)$ in T iterations. Such f^i that are exp-concave in u^i satisfying Assumption 2.1 arise in optimization under uncertainty problems where variance is used as a risk measure, e.g., mean-variance portfolio optimization problems, see for example [3, Example 25]. ∎

Our modification of the pessimization oracle-based approach of [26] requires $r_x(\epsilon)$ number of calls to the pessimization oracles (19). Specifically, Theorem 4.1 implies that the number of iterations required to solve the robust feasibility problem (4) with pessimization oracles (19) is the number of iterations T needed to guarantee $\epsilon^\bullet(\{x_t, u_t, \theta_t\}_{t=1}^T) \le \epsilon/2$. As before, this can be done via cheap OFO-based algorithms (see Theorem 2.1) requiring at most $O(1/\epsilon^2)$ iterations. Taking into account that our binary search factor $O(\log(1/\epsilon))$ to find a robust ϵ-optimal solution, the total number of iterations required is $O(\log(1/\epsilon)/\epsilon^2)$, which is much better than the exponential $(1+O(1/\epsilon))^n$ bound of [26, Section 5.2] that uses a full nominal solution oracle (18). Also, the per-iteration arithmetic cost involves calling m pessimization oracles (19) and a solution oracle. If $\sup_{u^i \in U^i} f^i(x,u^i)$ has a simple closed form solution, then the resulting arithmetic cost is $O(k)$ for each pessimization oracle. If we can use polynomial-time IPMs, this cost becomes $O(k^3 \log(1/\epsilon))$ (see [7, Section 6.6]), and using FOMs has cost $O(k \log(1/\epsilon))$ in the best case when the f^i are smooth *and* strongly convex in u^i.

The nominal feasibility/optimization oracle-based approach of [5] requires $r_u(\epsilon)$ number of calls

Table 1: Summary of different approaches to generate $\{x_t, u_t\}_{t=1}^T$.

Approach	Binary search	No. iterations	Per-iteration cost
OFO-based	$\log(1/\epsilon)$	$\max\{r_u(\epsilon), r_x(\epsilon)\}$	$O(km + mn)$
Pessimization oracle type:			
Closed form	$\log(1/\epsilon)$	$r_x(\epsilon)$	$O(km + mn)$
IPM	$\log(1/\epsilon)$	$r_x(\epsilon)$	$O(k^3 m \log(1/\epsilon) + mn)$
FOM (when f^i are smooth, str. conv. in u^i)	$\log(1/\epsilon)$	$r_x(\epsilon)$	$O\left(km \log(1/\epsilon) + mn\right)$
Feasibility oracle type:			
IPM	1	$r_u(\epsilon)$	$O\left(km + \sqrt{m}(n^3 + mn)\log(1/\epsilon)\right)$
CoMirror	1	$r_u(\epsilon)$	$O\left(km + mn/\epsilon^2\right)$
Convex-concave SP (when f^i are smooth in x)	$\log(1/\epsilon)$	$r_u(\epsilon)$	$O\left(km + \log(m)mn/\epsilon\right)$

to the nominal optimization oracle (or $r_u(\epsilon)\log(1/\epsilon)$ calls to the nominal feasibility oracle). Then their overall convergence rate in terms of basic arithmetic operations depends on the type of solver used to solve the nominal optimization/feasibility problem (20). We next examine the cost of implementing these oracles.

When applicable, polynomial-time IPMs are guaranteed to terminate in $O(\sqrt{m}\log(1/\epsilon))$ iterations with a solution to (20) and thus offer the best rates in terms of their dependence on ϵ. They also have the advantage that they can act as a nominal optimization oracle, and hence by Remark 4.5 there will be no need to perform an additional binary search to find an ϵ-optimal solution. On the other hand, they demand significantly more memory, and their per-iteration cost is quite high in terms of the dimension, usually around the order of $O(n^3 + mn)$, see [7, Chapter 6.6]. In order to keep both the memory requirements and the per-iteration cost associated with implementing the nominal feasibility oracle low, we may opt for an FOM called CoMirror algorithm, see [1] and [22, Section 1.3]. CoMirror algorithm is guaranteed to find a solution to the nominal ϵ-feasibility problem within $O(1/\epsilon^2)$ iterations, with a much cheaper per-iteration cost of $O(mn)$. Because CoMirror method can optimize as well, it does not need to binary search. However, it cannot exploit further structural properties of the functions f^i, such as smoothness in x, to improve the dependence on ϵ. In order to exploit such properties, it is possible to cast (20) as a convex-concave SP problem, and then apply efficient FOMs such as Nesterov's algorithm [29] or Nemirovski's Mirror Prox algorithm [27] to achieve a convergence rate of $O(\log(m)/\epsilon)$ and per-iteration cost of $O(mn)$. This convex-concave SP approach can only be used as a nominal feasibility oracle, so we must repeat the process $\log(1/\epsilon)$ times to obtain an ϵ-optimal solution.

In Table 1, we summarize the rates for the various approaches. Note that the total *arithmetic complexity* of each approach is obtained by multiplying the quantities in each row. Furthermore, the quantities $r_u(\epsilon), r_x(\epsilon)$ will generally be $O(1/\epsilon^2)$, with potential for application-specific acceleration when the functions f^i exhibit favorable structure.

Table 1 indicates that our modified pessimization oracle-based approach when it admits a closed form solution and the nominal feasibility oracle-based approach which uses a polynomial-time IPM solver to implement the oracle give the best dependence on ϵ among all of the methods. These are better than our OFO-based approach by factors of $\max\{1, r_u(\epsilon)/r_x(\epsilon)\}$ and $\max\{1, r_x(\epsilon)/r_u(\epsilon)\}$ respectively. However, in many applications, we can expect that $r_u(\epsilon) \approx r_x(\epsilon)$, so these factors will be constant. In this case, our OFO-based approach becomes competitive with having a closed form pessimization oracle or using a nominal IPM solver in [5]. However, compared to IPMs, our approach demands much less memory, and it is able to maintain a much lower dependence on the dimensions m, n and thus is much more scalable, whereas the cost per iteration of such IPMs has a rather high dependence on the dimension. In addition, their memory requirements are far more than OFO algorithms, posing a critical disadvantage to their use in large-scale applications. Similar comparisons of our OFO-based approach against pessimization or nominal oracle-based approaches utilizing other methods point out its advantage, which is at least an order of magnitude better in terms of its dependence on ϵ. In fact, when $r_x(\epsilon) \approx r_u(\epsilon)$, our method can lead to savings over the approach of [5] with CoMirror algorithm used in its oracle by a factor as large as $O(1/(\epsilon^2 \log(1/\epsilon)))$.

5 Conclusion

In this paper, we advance the line of research in [12, 13, 26, 5] that aims to solve robust optimization problems via iterative techniques, i.e., without transforming them into their equivalent robust counterparts. Thus far, the literature on iterative methods for RO has relied on more expensive pessimization or nominal feasibility oracles. However, in many applications of robust convex optimization, the original deterministic problem comes equipped with first-order oracles that provide gradient/subgradient information on the constraint functions f^i. In this paper, we present an efficient framework that can both work with cheap online first-order oracles and also capture the prior oracle-based approaches of [26] and [5]. We further show that working with these OFO oracles essentially does not increase the number of overall oracle calls, i.e., the number of main iterations of our approach is better than or comparable to the prior approaches. Moreover, when OFO oracles are utilized in our framework, the resulting overall arithmetic complexity including all of the basic operations in each iteration is remarkably cheaper than the prior approaches. The resulting framework is simple, easy-to-implement, flexible, and it can easily be customized to many applications. We demonstrate our framework via an illustrative robust QP example, where the most expensive operation in each iteration of our framework is a maximum eigenvalue computation.

Our framework is amenable to exploiting favorable structural properties of the functions f^i such as strong concavity, smoothness, etc., through which better convergence rates can be achieved. For example, when f^i are strongly concave in u^i, by exploiting this structural information and using a customization of the weighted regret online mirror descent for strongly convex functions, it is possible to achieve a better convergence rate of $O(1/\epsilon)$ in both our online first-order oracle setup and the nominal feasibility oracle framework of [5]. This then partially resolves/refines an open question stated in [5] for the lower bound on the number of iterations/calls needed in their nominal feasibility oracle based framework. However, it remains open whether $O(1/\epsilon^2)$ bound is tight when no further favorable structure is present in f^i or the tightness of $O(1/\epsilon)$ in the favorable case.

There are several other compelling avenues for future research. From a practical perspective, it is well-known, and also confirmed by our preliminary proof-of-concept computational experiments, that the computation of gradients/subgradients constitute a major bottleneck in the practical performance of FOMs. Thus, as a step to reduce the efforts involved in such computations, possible incorporation of stochastic [31, 28] and/or randomized FOMs [21, 10] working with stochastic subgradients into our framework is of great practical and theoretical interest. A critical assumption in our approach as well as others, e.g., see [5] and references therein, is that the domain X is convex. Removing the convexity requirement on the domain X will be an important theoretical development on its own. Besides, this will open up possibilities for more principled approaches to solving robust combinatorial optimization problems (see [11, 12]) where such a convexity assumption on X is not satisfied. Finally, another attractive research direction is develop analogous frameworks for multi-stage RO problems such as robust Markov decision processes (see [30, 19]).

Acknowledgments

This research is supported in part by NSF grant CMMI 1454548.

The authors would like to thank Arkadi Nemirovski for suggesting the convex-concave reformulation of the robust feasibility problem presented in Section A, in particular Lemma A.2.

References

[1] A. Beck, A. Ben-Tal, N. Guttmann-Beck, and L. Tetruashvili. The CoMirror algorithm for solving nonsmooth constrained convex problems. *Operations Research Letters*, 38(6):493 – 498, 2010.

[2] A. Ben-Tal, S. Bhadra, C. Bhattacharyya, and A. Nemirovski. Efficient methods for robust classification under uncertainty in kernel matrices. *Journal of Machine Learning Research*, 13(1):2923–2954, Oct. 2012.

[3] A. Ben-Tal, D. den Hertog, and J.-P. Vial. Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1-2):265–299, 2015.

[4] A. Ben-Tal, L. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[5] A. Ben-Tal, E. Hazan, T. Koren, and S. Mannor. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015.

[6] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[7] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001.

[8] A. Ben-Tal and A. Nemirovski. Robust optimization – methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

[9] A. Ben-Tal and A. Nemirovski. Selected topics in robust convex optimization. *Mathematical Programming*, 112(1):125–158, 2008.

[10] A. Ben-Tal and A. Nemirovski. On solving large-scale polynomial convex problems by randomized first-order algorithms. *Mathematics of Operations Research*, 40(2):474–494, 2015.

[11] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[12] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71, 2003.

[13] G. Calafiore and M. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[14] C. Caramanis, S. Mannor, and H. Xu. Robust optimization in machine learning. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2012.

[15] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[16] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[17] E. Hazan. The convex optimization approach to regret minimization. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2012.

[18] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[19] G. N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[20] V. Jeyakumar and G. Y. Li. Trust-region problems with linear inequality constraints: Exact SDP relaxation, global optimality and robust optimization. *Mathematical Programming*, 147(1):171–206, 2013.

[21] A. Juditsky, F. Kılınç-Karzan, and A. Nemirovski. Randomized first order algorithms with applications to ℓ_1 minimization. *Mathematical Programming*, 142(1-2):269–310, 2013.

[22] A. Juditsky and A. Nemirovski. First-order methods for nonsmooth convex large-scale optimization, I: General purpose methods. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2012.

[23] A. Juditsky and A. Nemirovski. First-order methods for nonsmooth convex large-scale optimization, II: Utilizing problem's structure. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2012.

[24] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari. Regularization techniques for learning with matrices. *Journal of Machine Learning Research*, 13(Jun):1865–1890, 2012.

[25] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291 – 307, October 2005.

[26] A. Mutapcic and S. Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods and Software*, 24(3):381–406, June 2009.

[27] A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[28] A. S. Nemirovski, A. B. Juditsky, G. Lan, and A. Shapiro. Stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[29] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[30] A. Nilim and L. El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[31] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[32] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[33] P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7(Jul):1283–1314, 2006.

[34] H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(Jul):1485–1510, 2009.

[35] H. Xu, C. Caramanis, and S. Mannor. Robust regression and Lasso. *IEEE Transactions on Information Theory*, 56(7):3561–3574, July 2010.

[36] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 928–936, 2003.

A Convex-Concave Saddle Point Reformulation

The SP problem (7) based on the function $\Phi(x, u)$ which is not necessarily concave in u admits a convex-concave SP representation in a lifted space via perspective transformations. To present this reformulation, we start by defining the following sets with additional variables $y \in \mathbb{R}^m_+$ and new variables v^i for $i \in [m]$:

$$V^i = \left\{ [v^i; y^{(i)}] : \ 0 < y^{(i)} \leq 1, \ \frac{v^i}{y^{(i)}} \in U^i \right\} \quad \forall i \in [m],$$
$$W = \left\{ w = [v^1; \ldots; v^m; y] : \ [v^i; y^{(i)}] \in \mathrm{cl}(V^i), \ i \in [m], \ \sum_{i=1}^m y^{(i)} = 1 \right\}.$$

Note that for all $i \in [m]$, $\mathrm{cl}(V^i) = V^i \cup \{[0; 0]\}$ because we assumed U^i to be closed sets. For the point $[v^i; y^{(i)}] = [0; 0]$, we set $y^{(i)} f^i \left(x, \frac{v^i}{y^{(i)}} \right) = 0$ for any $x \in X$. Note that setting $y^{(i)} f^i \left(x, \frac{v^i}{y^{(i)}} \right) = 0$ for $[v^i; y^{(i)}] = [0; 0]$ is well-defined as the continuation since from Assumption 2.1, $f^i(x, u^i)$ is continuous and finite-valued on U^i, and U^i is compact, so we can deduce that $f^i(x, u^i)$ will be bounded on U^i.

We also define the function $\psi : X \times W \to \mathbb{R}$ as

$$\psi(x, w) = \psi(x, v, y) := \sum_{i=1}^m y^{(i)} f^i \left(x, \frac{v^i}{y^{(i)}} \right).$$

Lemma A.1. *For fixed $w \in W$, the function $\psi(x, w)$ is convex in x over X, and $\psi(x, w)$ is a concave function of w over W for any fixed x. Moreover, W is closed, and when U^i for $i \in [m]$ are convex, the sets V^i for $i \in [m]$ and W are all convex.*

Proof. For any $w = [v^1; \ldots; v^m; y]$, the function ψ is convex in x since in all of the nonzero terms in the summation over all $i \in [m]$ defining ψ, we have $y^{(i)} > 0$ and in each such nonzero term each function $f^i \left(x, \frac{v^i}{y^{(i)}} \right)$ is convex in x for the given $\frac{v^i}{y^{(i)}} \in U^i$ (see Assumption 2.1). In addition, for any given $x \in X$, the function ψ is jointly concave in v and y because it is written as a sum of the perspective functions of functions f^i which are concave in u^i (see Assumption 2.1).

The closedness of W is immediate, and the convexity of the sets V^i and W follows immediately from their definition and the convexity assumption on U^i. □

With these definitions and Lemma A.1, we observe that (7) is equivalent to evaluating the convex-concave SP problem defined by the function ψ over the convex domains X and W:

$$\inf_{x \in X} \sup_{w \in W} \psi(x, w) \leq \epsilon \quad \text{or} \quad \inf_{x \in X} \sup_{w \in W} \psi(x, w) > 0. \tag{22}$$

We state this formally in the following lemma.

Lemma A.2. *For any $\epsilon > 0$ and $\bar{x} \in X$,*

$$\max_{i \in [m]} \sup_{u^i \in U^i} f^i(\bar{x}, u^i) \leq \epsilon \quad \textit{if and only if} \quad \sup_{w \in W} \psi(\bar{x}, w) \leq \epsilon.$$

As a result,

$$\inf_{x \in X} \max_{i \in [m]} \sup_{u^i \in U^i} f^i(x, u^i) \leq \epsilon \quad \textit{if and only if} \quad \inf_{x \in X} \sup_{w \in W} \psi(x, w) \leq \epsilon.$$

Proof. Fix $\bar{x} \in X$ and $\epsilon > 0$. Suppose $\max_{i\in[m]} \sup_{u^i \in U^i} f^i(\bar{x}, u^i) \le \epsilon$; then for all $u^i \in U^i$, $i \in [m]$, we have $f^i(\bar{x}, u^i) \le \epsilon$. Now consider any $w = [v^1; \dots; v^m; y] \in W$. Then $0 \le y^{(i)} \le 1$ for all $i \in [m]$ and $\sum_{i=1}^m y^{(i)} = 1$. For all $i \in [m]$, define $u^i = \frac{v^i}{y^{(i)}} \in U^i$ whenever $y^{(i)} > 0$. Then $y^{(i)} f^i(\bar{x}, \frac{v^i}{y^{(i)}}) = y^{(i)} f^i(\bar{x}, u^i) \le y^{(i)}\epsilon$ for $0 < y^{(i)} \le 1$. In addition, when $y^{(i)} = 0$, because $w \in W$ we must have $v^i = 0$ and then by definition we have $y^{(i)} f^i(\bar{x}, \frac{v^i}{y^{(i)}}) = 0$. Therefore, from $\sum_{i=1}^m y^{(i)} = 1$, we deduce $\psi(\bar{x}, w) = \sum_{i=1}^m y^{(i)} f^i\left(\bar{x}, \frac{v^i}{y^{(i)}}\right) \le \epsilon$ holds for any $w \in W$.

Now suppose that $\sup_{w\in W} \psi(\bar{x}, w) \le \epsilon$ holds. Given $i \in [m]$ and $u^i \in U^i$, set w to have components $y^{(i)} = 1$, $v^i = u^i$, and $[v^j; y^{(j)}] = [0; 0]$ for $j \ne i$. Then $f^i(\bar{x}, u^i) = \psi(\bar{x}, w) \le \epsilon$. Hence, $\max_{i\in[m]} \sup_{u^i\in U^i} f^i(\bar{x}, u^i) \le \epsilon$ follows. □

Remark A.1. When $m = 1$, i.e., we have only one function $f^1(x, u^1)$ and only one uncertainty set U^1, hence $W = U^1$ and $\inf_{x\in X} \sup_{v\in W} \psi(x, w) = \inf_{x\in X} \sup_{u^1\in U^1} f^1(x, u^1)$. Also, under Assumption 2.1, $\psi(x, w)$ is convex in x and concave in u^1. Thus, the preceding perspective transformation resulting in (22) directly generalizes this case of a convex-concave SP formulation for $m = 1$ discussed in Remark 3.1. ∎

As a result, Lemma A.2 and Theorem 3.1 combined with any FOM that provides bounds on the saddle point gap $\epsilon^{\psi}_{\text{sad}}(\bar{x}, \bar{w})$ lead to an efficient way of verifying robust feasibility of (7) as follows:

Theorem A.1. *Suppose $\bar{x} \in X$, $\bar{w} \in W$, and $\tau \in (0, 1)$ are such that $\epsilon^{\psi}_{\text{sad}}(\bar{x}, \bar{w}) \le \tau\epsilon$. If $\psi(\bar{x}, \bar{w}) \le (1 - \tau)\epsilon$, then $\max_{i\in[m]} \sup_{u^i\in U^i} f^i(\bar{x}, u^i) \le \epsilon$. If $\psi(\bar{x}, \bar{w}) > (1 - \tau)\epsilon$ and $\tau \le \frac{1}{2}$, then $\inf_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) > 0$.*

Proof. Suppose $\psi(\bar{x}, \bar{w}) \le \tau\epsilon$. By Theorem 3.1, we have $\inf_{x\in X} \sup_{w\in W} \psi(x, w) \le \sup_{w\in W} \psi(\bar{x}, w) \le \epsilon$. By Lemma A.2, $\max_{i\in[m]} \sup_{u^i\in U^i} f^i(\bar{x}, u^i) \le \epsilon$ as well.

On the other hand, when $\psi(\bar{x}, \bar{w}) > (1 - \tau)\epsilon$ and $\tau \le \frac{1}{2}$, Theorem 3.1 implies $\inf_{x\in X} \sup_{w\in W} \psi(x, w) \ge \inf_{x\in X} \psi(x, \bar{w}) > 0$. Then by Lemma A.2, $\max_{i\in[m]} \sup_{u^i\in U^i} f^i(\bar{x}, u^i) > 0$ follows. □

Because of the existence of efficient FOMs to solve convex-concave SP problems, Theorem A.1 suggests a possible advantage of using the convex-concave SP problem given in (22). Nevertheless, working with the SP reformulation given by (22) in the extended space $X \times W$ presents a number of critical challenges. First, efficient FOMs associated with convex-concave SP problems often require computing prox operations or projections onto the domains X and W. Unfortunately, even if projection (or prox-mappings) onto U^i admits a closed form solution or an efficient procedure, it is unclear how to extend such projections onto W. Furthermore, while the perspective transformations involved in constructing the function ψ preserves certain desirable properties of the functions f^i, such as Lipschitz continuity and smoothness, the parameters associated with ψ are in general larger than those associated with the original functions f^i. Such parameters are critical for FOM convergence rates, and thus the FOMs when applied to solve (22) will have slower convergence rates.

To address the issues outlined above, in the main paper we discuss how to obtain robust feasibility/infeasibility certificates for the convex-nonconcave SP problem (7) directly, i.e., we work with the functions f^i and the sets U^i directly. This direct approach in particular allows us to take greater advantage of the structure of the original formulation such as the availability of efficient projection (prox) computations over domains, and/or better parameters for smoothness, Lipschitz continuity, etc., of the functions.