We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

BOOK
CITATION
INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Distributionally Robust Optimization

Jian Gao, Yida Xu, Julian Barreiro-Gomez,
Massa Ndong, Michalis Smyrnakis and
Hamidou Tembine

Additional information is available at the end of the chapter

Abstract

This chapter presents a class of distributionally robust optimization problems in which a decision-maker has to choose an action in an uncertain environment. The decision-maker has a continuous action space and aims to learn her optimal strategy. The true distribution of the uncertainty is unknown to the decision-maker. This chapter provides alternative ways to select a distribution based on empirical observations of the decision-maker. This leads to a distributionally robust optimization problem. Simple algorithms, whose dynamics are inspired from the gradient flows, are proposed to find local optima. The method is extended to a class of optimization problems with orthogonal constraints and coupled constraints over the simplex set and polytopes. The designed dynamics do not use the projection operator and are able to satisfy both upper- and lower-bound constraints. The convergence rate of the algorithm to generalized evolutionarily stable strategy is derived using a mean regret estimate. Illustrative examples are provided.

Keywords: distribution robustness, gradient flow, Bregman divergence, Wasserstein metric, f-divergence

1. Introduction

Robust optimization can be defined as the process of determining the best or most effective result, utilizing a quantitative measurement system under worst case uncertain functions or parameters. The optimization may occur in terms of best robust design, net cash flows, profits, costs, benefit/cost ratio, quality-of-experience, satisfaction, end-to-end delay, completion time, etc. Other measurement units may be used, such as units of production or production time, and optimization may occur in terms of maximizing production units, minimizing processing time,

production time, maximizing profits, or minimizing costs under uncertain parameters. There are numerous techniques of robust optimization methods such as robust linear programming, robust dynamic programming, robust geometric programming, queuing theory, risk analysis, etc. One of the main drawbacks of the robust optimization is that the worst scenario may be too conservative. The bounds provided by the worst case scenarios may not be useful in many interesting problems (see the wireless communication example provided below). However, distributionally robust optimization is not based on the worst case parameters. The distributional robustness method is based the probability distribution instead of worst parameters. The worse case distribution within a certain carefully designed distributional uncertainty set may provide interesting features. Distributionally robust programming can be used not only to provide a distributionally robust solution to a problem when the true distribution is unknown, but it also can, in many instances, give a general solution taking into account some risk. The presented methodology is simple and reduces significantly the dimensionality of the distributionally robust optimization. We hope that the designs of distributionally robust programming presented here can help designers, engineers, cost–benefit analyst, managers to solve concrete problems under unknown distribution.

The rest of the chapter is organized as follows. Section 2 presents some preliminary concepts of distributionally robust optimization. A class of constrained distributionally robust optimization problems are presented in Section 3. Section 4 focuses on distributed distributionally robust optimization. Afterwards, illustrative examples in distributed power networks and in wireless communication networks are provided to evaluate the performance of the method. Finally, prior works and concluding remarks are drawn in Section 5.

Notation: Let $\mathbb{R}$, $\mathbb{R}_+$, denote the set of real and non-negative real numbers, respectively, (Ω, d) be a separable completely metrizable topological space with $d : \Omega \times \Omega \to \mathbb{R}_+$ a metric (distance). Let $\mathcal{P}(\Omega)$ be the set of all probability measures over Ω.

2. Distributionally robust optimization

This section introduces distributionally robust optimization models. We will first present a generic formulation of the problem. Then, individual components of the optimization and their solvability issues via equivalent formulations will be discussed.

2.1. Model

Consider a decision-maker who wants to select an action $a \in \mathcal{A} \subset \mathbb{R}^n$ in order to optimize her objective $r(a, \omega)$, where ω is an uncertain parameter. The information structure is the following:

- The true distribution of ω is not known to the decision-maker.

- The upper/lower bound (if any) of ω are unknown to the decision-maker.

- The decision-maker can measure/observe realization of the random variable ω.

The decision-maker chooses to experiment several trials and obtains statistical realizations of ω from measurements. The measurement data can be noisy, imperfect and erroneous. Then, an empirical distribution (or histogram) m is built from the realizations of ω. However, m is not the true distribution of the random variable ω, and m may not be a reliable measure due to statistical, bias, measurement, observation or computational errors. Therefore, the decision-maker is facing a risk. The risk-sensitive decision-maker should decide action that improves the performance of $\mathbb{E}_{\tilde{m}} r(a, \omega)$ among alternative distributions $\tilde{m}$ within a certain level of deviation $\rho > 0$ from the distribution m. The distributionally robust optimization problem is therefore formulated as

$$\sup_{a \in \mathcal{A}} \inf_{\tilde{m} \in B_\rho(m)} \mathbb{E}_{\omega \sim \tilde{m}} r(a, \omega). \tag{1}$$

where $B_\rho(m)$ is the uncertainty set of alternative admissible distributions from m within a certain radius $\rho > 0$. Different distributional uncertainty sets are presented: the f-divergence and the Wasserstein metric, defined below.

2.1.1. f-divergence

We introduce the notion of $f-$ divergence which will be used to compute the discrepancy between probability distributions.

Definition 1. *Let m and $\tilde{m}$ be two probability measures over Ω such that m is absolutely continuous with respect to $\tilde{m}$. Let f be a convex function. Then, the f-divergence between m and $\tilde{m}$ is defined as follows:*

$$D_f(m\|\tilde{m}) \equiv \int_\Omega f\left(\frac{dm}{d\tilde{m}}\right) d\tilde{m} - f(1),$$

where $\frac{dm}{d\tilde{m}}$ is the Radon-Nikodym derivative of the measure m with the respect the measure $\tilde{m}$.

By Jensen's inequality:

$$\begin{aligned} D_f(m\|\tilde{m}) &= \int_\Omega f\left(\frac{dm}{d\tilde{m}}\right) d\tilde{m} - f(1) \\ &\geq f\left(\int_\Omega \frac{dm}{d\tilde{m}} d\tilde{m}\right) - f(1) \\ &= f\left(\int_\Omega dm\right) - f(1) \\ &= f(1) - f(1) = 0. \end{aligned} \tag{2}$$

Thus, $D_f(m\|\tilde{m}) \geq 0$ for any convex function f. Note however that, the $f-$ divergence $D_f(m\|\tilde{m})$ is not a distance (for example, it does not satisfy the symmetry property). Here the distributional uncertainty set imposed to the alternative distribution $\tilde{m}$ is given by

$$B_\rho(m) = \left\{ \tilde{m} \,|\, \tilde{m}(.) \geq 0, \ \int_\Omega d\tilde{m} = \tilde{m}(\Omega) = 1, \ D_f(\tilde{m}\|m) \leq \rho \right\}.$$

Example 1. *From the notion of* $f-$ *divergence one can derive the following important concept:*

- *α-divergence for*

$$f(a) = \begin{cases} \dfrac{4}{(\alpha+1)(1-\alpha)} \left(1 - a^{\frac{\alpha+1}{2}}\right) & \text{if} \ \ \alpha \notin \{-1, +1\}, \\ a\log a & \text{if} \ \ \alpha = 1, \\ -\log a & \text{if} \ \ \alpha = -1, \end{cases}$$

- *In particular, Kullback–Leibler divergence (or relative entropy) is retrieved as α goes to 1.*

2.1.2. Wasserstein metric

The Wasserstein metric between two probability distributions $\tilde{m}$ and m is defined as follows:

Definition 2. *For* $m, \tilde{m} \in \mathcal{P}(\Omega)$, *let* $\Pi(\tilde{m}; m)$ *be the set of all couplings between m and $\tilde{m}$. That is,*

$$\left\{ \pi \in \mathcal{P}(\Omega \times \Omega) \ | \ \pi(A \times \Omega) = m(A), \ \pi(\Omega \times B) = \tilde{m}(B), \ (A, B) \in \mathcal{B}^2(\Omega) \right\}.$$

$\mathcal{B}(\Omega)$ denotes the measurable sets of Ω. Let $\theta \in [1, \infty]$. The Wasserstein metric between $\tilde{m}$ and m is defined as

$$W_\theta(\tilde{m}; m) = \inf_{\pi \in \Pi(\tilde{m};m)} \|d\|_{L^\theta_\pi} = \inf_{\pi \in \Pi(\tilde{m};m)} \int_{(a,b)} d^\theta(a, b)\pi(da, db),$$

It is well-known that for every $\theta \geq 1$, $W_\theta(\tilde{m}; m)$ is a true distance in the sense that it satisfies the following three axioms:

- positive-definiteness,

- the symmetry property,

- the triangle inequality.

Note that $\tilde{m}$ is not necessarily absolutely continuous with respect to m. Now the distributional uncertainty/constraint set is the set of all possible probability distributions within a L^θ-Wasserstein distance below ρ.

$$\tilde{B}_\rho(m) = \left\{ \tilde{m} \,|\, \int_\Omega d\tilde{m} = \tilde{m}(\Omega) = 1, \ W_\theta(\tilde{m}; m) \leq \rho \right\},$$

Note that, if m is a random measure (obtained from a sampled realization), we use the expected value of the Wasserstein metric.

Example 2. *The L^θ-Wasserstein distance between two Dirac measures δ_{ω_0} and $\delta_{\tilde{\omega}_0}$ is $W_\theta(\delta_{\omega_0}, \delta_{\tilde{\omega}_0}) = d(\omega_0, \tilde{\omega}_o)$. More generally, for $K \geq 2$, the L^2-Wasserstein distance between empirical measures $\mu_K = \frac{1}{K}\sum_{k=1}^K \delta_{\omega_k}$ and $\nu_K = \frac{1}{K}\sum_{k=1}^K \delta_{\tilde{\omega}_k}$ is $W_2^2(\mu_K, \nu_K) \leq \frac{1}{K}\sum_{i=1}^K [\omega_k - \tilde{\omega}_k]^2$.*

We have defined $B_\rho(m)$ and $\tilde{B}_\rho(m)$. The goal now is to solve (1) under both $f-$ divergence and Wasserstein metric. One of the difficulties of the problem is the curse of dimensionality. The distributionally robust optimization problem (1) of the decision-maker is an infinite-dimensional robust optimization problem because B_ρ is of infinite dimensions. Below we will show that (1) can be transformed into an optimization in the form of supinfsup. The latter problem has three alternating terms. Solving this problem requires a triality theory.

2.2. Triality theory

We first present the duality gap and develop a triality theory to solve equivalent formulations of (1). Consider uncoupled domains $\mathcal{A}_i, i \in \{1, 2, 3\}$. For a general function r_2, one has

$$\sup_{a_2 \in \mathcal{A}_2} \inf_{a_1 \in \mathcal{A}_1} r_2(a_1, a_2) \leq \inf_{a_1 \in \mathcal{A}_1} \sup_{a_2 \in \mathcal{A}_2} r_2(a_1, a_2)$$

and the difference

$$\min_{a_1 \in \mathcal{A}_1} \max_{a_2 \in \mathcal{A}_2} r_2(a_1, a_2) - \max_{a_2 \in \mathcal{A}_2} \min_{a_1 \in \mathcal{A}_1} r_2(a_1, a_2),$$

is called duality gap. As it is widely known in duality theory from Sion's Theorem [1] (which is an extension of von Neumann minimax Theorem) the duality gap vanishes, for example for convex-concave function, and the value is achieved by a saddle point in the case of non-empty convex compact domain.

Triality theory focuses on optimization problems of the forms: sup infsup or infsup inf. The term triality is used here because there are three key alternating terms in these optimizations.

Proposition 1. *Let $(a_1, a_2, a_3) \mapsto r_3(a_1, a_2, a_3) \in \mathbb{R}$ be a function defined on the product space $\prod_{i=1}^3 \mathcal{A}_i$. Then, the following inequalities hold:*

$$\sup_{a_2 \in \mathcal{A}_2} \inf_{a_1 \in \mathcal{A}_1, a_3 \in \mathcal{A}_3} r_3(a_1, a_2, a_3) \leq$$
$$\inf_{a_3 \in \mathcal{A}_3} \sup_{a_2 \in \mathcal{A}_2} \inf_{a_1 \in \mathcal{A}_1} r_3(a_1, a_2, a_3) \leq \tag{3}$$
$$\inf_{a_1 \in \mathcal{A}_1, a_3 \in \mathcal{A}_3} \sup_{a_2 \in \mathcal{A}_2} r_3(a_1, a_2, a_3),$$

and similarly

$$\sup_{a_1 \in \mathcal{A}_1, a_3 \in \mathcal{A}_3} \inf_{a_2 \in \mathcal{A}_2} r_3(a_1, a_2, a_3) \leq$$
$$\sup_{a_3 \in \mathcal{A}_3} \inf_{a_2 \in \mathcal{A}_2} \sup_{a_1 \in \mathcal{A}_1} r_3(a_1, a_2, a_3) \leq \tag{4}$$
$$\inf_{a_2 \in \mathcal{A}_2} \sup_{a_1 \in \mathcal{A}_1, a_3 \in \mathcal{A}_3} r_3(a_1, a_2, a_3).$$

Proof. Define

$$\hat{g}(a_2, a_3) := \inf_{a_1 \in \mathcal{A}_1} r_3(a_1, a_2, a_3).$$

Thus, for all a_2, a_3, one has $\hat{g}(a_2, a_3) \le r_3(a_1, a_2, a_3)$. It follows that, for any a_1, a_3,

$$\sup_{a_2 \in \mathcal{A}_2} \hat{g}(a_2, a_3) \le \sup_{a_2 \in \mathcal{A}_2} r_3(a_1, a_2, a_3).$$

Using the definition of $\hat{g}$, one obtains

$$\sup_{a_2 \in \mathcal{A}_2} \inf_{a_1 \in \mathcal{A}_1} r_3(a_1, a_2, a_3) \le \sup_{a_2 \in \mathcal{A}_2} r_3(a_1, a_2, a_3), \quad \forall a_1, a_3.$$

Taking the infimum in a_1 yields:

$$\sup_{a_2 \in \mathcal{A}_2} \inf_{a_1 \in \mathcal{A}_1} r_3(a_1, a_2, a_3) \le \inf_{a_1 \in \mathcal{A}_1} \sup_{a_2 \in \mathcal{A}_2} r_3(a_1, a_2, a_3), \quad \forall a_3 \tag{5}$$

Now, we use two operations for the variable a_3:

- Taking the infimum in the inequality (5) in a_3 yields

$$\inf_{a_3 \in \mathcal{A}_3} \sup_{a_2 \in \mathcal{A}_2} \inf_{a_1 \in \mathcal{A}_1} r_3(a_1, a_2, a_3) \le \inf_{a_3 \in \mathcal{A}_3} \inf_{a_1 \in \mathcal{A}_1} \sup_{a_2 \in \mathcal{A}_2} r_3(a_1, a_2, a_3)$$

$$= \inf_{(a_1, a_3) \in \mathcal{A}_1 \times \mathcal{A}_3} \sup_{a_2 \in \mathcal{A}_2} r_3(a_1, a_2, a_3),$$

which proves the second part of the inequalities (3). The first part of the inequalities (3) follows immediately from (5).

- Taking the supremum in inequality (5) in a_3 yields

$$\sup_{(a_2, a_3) \in \mathcal{A}_2 \times \mathcal{A}_3} \inf_{a_1 \in \mathcal{A}_1} r_3(a_1, a_2, a_3) \le \sup_{a_3 \in \mathcal{A}_3} \inf_{a_1 \in \mathcal{A}_1} \sup_{a_2 \in \mathcal{A}_2} r_3(a_1, a_2, a_3),$$

which proves the first part of the inequalities (4). The second part of the inequalities (4) follows immediately from (5).

This completes the proof.

2.3. Equivalent formulations

Below we explain how the dimensionality of problem (1) can be significantly reduced using a representation by means of the triality theory inequalities of Proposition 1.

2.3.1. *f-divergence*

Interestingly, the distributionally robust optimization problem (1) under *f*-divergence is equivalent to the finite dimensional stochastic optimization problem (when $\mathcal{A}$ are of finite

dimensions). To see this, the original problem need to be transformed. Let us introduce the likelihood functional $L(\tilde{\omega}) = \frac{d\tilde{m}}{dm}(\tilde{\omega})$, and set

$$L_\rho(m) = \left\{ L | \int_{\tilde{\omega}} f(L(\tilde{\omega}))dm - f(1) \le \rho, \quad \int_{\tilde{\omega}} L(\tilde{\omega})dm(\tilde{\omega}) = 1 \right\}.$$

Then, the Lagrangian of the problem is

$$\tilde{r}(a, L, \lambda, \mu) = \int_{\tilde{\omega}} r(a, \tilde{\omega})L(\tilde{\omega})dm(\tilde{\omega})$$

$$- \lambda \left(\rho + f(1) - \int_{\tilde{\omega}} f(L(\tilde{\omega}))dm(\tilde{\omega}) \right)$$

$$- \mu \left(1 - \int_{\tilde{\omega}} L(\tilde{\omega})dm(\tilde{\omega}) \right),$$

where $\lambda \ge 0$ and $\mu \in \mathbb{R}$. The problem becomes

$$\left\{ \sup_a \inf_{L \in L_\rho(m)} \sup_{\lambda \ge 0, \mu \in \mathbb{R}} \tilde{r}(a, L, \lambda, \mu). \right. \tag{6}$$

A full understanding of problem (6) requires a triality theory (not a duality theory). The use of triality theory leads to the following equation:

$$\left\{ \sup_{a \in \mathcal{A}} \inf_{\tilde{m} \in B_\rho(m)} \mathbb{E}_{\tilde{m}}[r] = \sup_{a \in \mathcal{A}, \lambda \ge 0, \mu \in \mathbb{R}} \mathbb{E}_m h, \right. \tag{7}$$

where h is the integrand function $-\lambda(\rho + f(1)) - \mu - \lambda f^* \left(\frac{r+\mu}{-\lambda} \right)$, where f^* is Legendre-Fenchel transform of f defined by

$$f^*(\xi) = \sup_L [\langle L, \xi \rangle - f(L)] = - \inf_L [f(L) - \langle L, \xi \rangle]. \tag{8}$$

Note that the righthand side of (7) is of dimension $n + 2$, which reduces considerably the dimensionality of the original problem (1).

2.3.2. Wasserstein metric

Similarly, the distributionally robust optimization problem under Wasserstein metric is equivalent to the finite dimensional stochastic optimization problem (when $\mathcal{A}$ is a set of finite dimension). If the function $\omega \mapsto r(a, \omega)$ is upper semi-continuous and (Ω, d) is a Polish space then the Wasserstein distributionally robust optimization problem is equivalent to

$$\left\{ \begin{array}{l} \sup_{a \in \mathcal{A}} \inf_{\tilde{m} \in \tilde{B}_\rho}(m) \mathbb{E}_{\tilde{m}}[r] = \sup_{a \in \mathcal{A}} \sup_{\lambda \ge 0} \mathbb{E}_m \left[\tilde{h} \right], \\ \tilde{h} = \lambda \rho^\theta + \mu + \sup_{\hat{\omega} \in \Omega} [r(a, \omega) - \mu - \lambda d^\theta(\omega, \hat{\omega})]; \end{array} \right. \tag{9}$$

The next subsection presents algorithms for computing a distributionally robust solution from the equivalent formulations above.

2.4. Learning algorithms

Learning algorithms are crucial for finding approximate solutions to optimization and control problems. They are widely used for seeking roots/kernel of a function and for finding feasible solutions to variational inequalities. Practically, a learning algorithm generates a certain trajectory (or a set of trajectories) toward a potential approximate solution. Selecting a learning algorithm that has specific properties such as better accuracy, more stability, less-oscillatory and quick convergence is a challenging task [2–5]. From the calculus of variations point of view, however, a learning algorithm generates curves. Therefore, selecting an algorithm among the others leads to an optimal control problem on the spaces of curves. Hence, it is natural to use optimal control theory to derive faster algorithms for a family of curves. Bergman-based algorithms and risk-aware version of it are introduced below to meet specific properties. We start by introducing the Bregman divergence.

Definition 3. *The Bregman divergence $d_g : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ is defined on a differentiable strictly convex function $g : \mathcal{A} \to \mathbb{R}$. For two points $(a, b) \in \mathcal{A}^2$, it measures the gap between $g(a)$ and the first-order Taylor expansion of g around a evaluated at b*

$$d_g(a, b) := g(a) - g(b) - \langle \nabla g(b), a - b \rangle.$$

Example 3. *From the Bregman divergence one gets other features by choosing specific functions g :*

- *If $g(a) = \sum_{i=1}^{n} a_i^2$ then the Bregman divergence $d_g(a, b) = \sum_{i=1}^{n} (a_i - b_i)^2$ is the squared standard Euclidean distance.*

- If $g(a) = \sum_{i=1}^{n} a_i \log a_i$ is defined on the relative interior of the simplex, i.e., $a \in \{b \mid b \in (0, 1)^n, \ \sum_{i=1}^{n} b_i = 1\}$ then the Bregman divergence $d_g(a, b) = \sum_{i=1}^{n} a_i \log \left(\frac{a_i}{b_i}\right)$, is the Kullback–Leibler divergence.

We are now ready to define algorithms for solving the righthand side of (7) and (9). One of the key approaches for error quantification of the algorithm with respect to the distributionally robust optimum is the so-called average regret. When the regret vanishes one gets close to a distributionally robust optimum.

Definition 4. *The average regret of an algorithm which generates the trajectory $a(t) = (\tilde{a}(t), \lambda(t), \mu(t))$ within $[t_0, T]$, $t_0 > 0$ is*

$$regret_T := \frac{1}{T - t_0} \int_{t_0}^{T} \left[\max_{b \in \mathcal{A} \times \mathbb{R}_+ \times \mathbb{R}} \mathbb{E}_m h(b, \omega) \right] - \mathbb{E}_m h(a(t), \omega) dt$$

2.4.1. Armijo gradient flow

Algorithm 1. *The Armijo's gradient pseudocode is as follows:*

1: **Procedure** ARMIJO GRADIENT $(a(0), \epsilon, T, g, m, h) \triangleright$ *The Armijo's gradient starting from $a(0)$ within $[0, T]$*

2:      $a \leftarrow a(0)$

3:      **while** regret > ε and t ≤ T **do** ▷ We have the answer if regret is 0

4:          Compute a(t) solution of (10)

5:          Compute regret_t

6:      **end while**

7:      **return** a(t), regret_t ▷ get a(t) and the regret

8: **end procedure**

Proposition 2. *Let* $a \mapsto \mathbb{E}_m h(a, \omega) : \; \mathbb{R}^{n+2} \to \mathbb{R}$ *be a concave function that has a unique global maximizer* a^*. *Assume that* a^* *be a feasible action profile, i.e.,* $a^* \in \mathcal{A}$. *Consider the continuous time analogue of the Armijo gradient flow [6], which is given by*

$$\frac{d}{dt}a(t) = \left[\nabla^2 g\right]^{-1} . \nabla_a \mathbb{E}_m h(a(t), \omega),$$

$$a(0) = a_0 \in \mathbb{R}^{n+2}, \tag{10}$$

where $a(0) = a_0$ is the initial point of the algorithm and g is a strictly convex function on a. Let $a(t)$ be the solution to (10).

Then the average regret within $[t_0, T]$, $t_0 > 0$ *is bounded above by*

$$regret_T := \frac{1}{T - t_0} \int_{t_0}^{T} \mathbb{E}_m[h(a^*, \omega) - h(a(t), \omega)]dt \le d_g(a^*, a_0) \frac{\log \frac{T}{t_0}}{T - t_0}.$$

Proof. Let

$$W(a(t)) = t\mathbb{E}_m[h(a^*, \omega) - h(a(t), \omega)] + d_g(a^*, a(t)),$$

where a is solution to (10). The function W is positive and $\frac{d}{dt}W = \mathbb{E}_m[h(a^*, \omega) - h(a(t), \omega)] - t\langle \mathbb{E}_m \nabla_a h(a, \omega), g_{aa}^{-1} \mathbb{E}_m \nabla_a h(a(t), \omega)\rangle + \frac{d}{dt}d_g(a^*, a(t))$. By concavity of $\mathbb{E}_m h(a, \omega)$ one has

$$\langle \mathbb{E}_m \nabla_a h(a, \omega), (a^* - a)\rangle \ge \mathbb{E}_m[h(a^*, \omega) - h(a, \omega)], \; \forall \; a.$$

On the other hand,

$$\frac{d}{dt}d_g(a^*, a(t)) = -\dot{a}g_a(a) - \langle g_{aa}\dot{a}, a - a^*\rangle + g_a\dot{a} \tag{11}$$

$$= -\langle g_{aa}\dot{a}, a - a^*\rangle = -\langle \mathbb{E}_m \nabla_a h(a, \omega), a^* - a\rangle.$$

Hence,

$$\frac{d}{dt}W \le \langle \mathbb{E}_m \nabla_a h(a, \omega), (a^* - a)\rangle$$

$$-t\langle \mathbb{E}_m \nabla_a h(a, \omega), g_{aa}^{-1} \mathbb{E}_m \nabla_a h(a, \omega)\rangle \tag{12}$$

$$-\langle \mathbb{E}_m \nabla_a h(a, \omega), a^* - a\rangle$$

$$= -t\langle \mathbb{E}_m \nabla_a h(a, \omega), g_{aa}^{-1} \mathbb{E}_m \nabla_a h(a, \omega)\rangle \le 0,$$

where the last inequality is by convexity of g. It follows that $\frac{d}{dt}W(a(t)) \leq 0$ along the path of the gradient flow. This decreasing property implies $0 \leq W(a(t)) \leq W(a(0)) = d_g(a^*, a(0))$. In particular, $0 \leq t\mathbb{E}_m[h(a^*, \omega) - h(a, \omega)] \leq W(a(0)) < +\infty$. Thus, the error to the value $\mathbb{E}_m h(a^*, \omega)$ is bounded by

$$0 \leq \mathbb{E}_m[h(a^*, \omega) - h(a, \omega)] \leq \frac{W(a(0))}{t}.$$

The announced result on the regret follows by integration over $[t_0, T]$ and by averaging. This completes the proof.

Note that the above regret-bound is established without assuming strong convexity of $a \mapsto -\mathbb{E}_m h(a, \omega)$. Also no Lipschitz continuity bound of the gradient is assumed.

2.4.2. Bregman learning algorithms

Algorithm 2. *The Bregman learning pseudocode is as follows:*

*1: **procedure** BREGMAN $(a(0), \epsilon, T, g, \alpha, \beta, m, h) \triangleright$ The Bregman learning starting from $a(0)$ within $[0, T]$*

2: $a \leftarrow a(0)$

*3:   **while** regret $> \epsilon$ and $t \leq T$ **do** $\triangleright$ We have the answer if regret is 0*

4:   Compute $a(t)$ solution of (13)

5:   Compute $regret_t$

*6:   **end while***

*7:   **return** $a(t), regret_t \triangleright$ get $a(t)$ and the regret*

*8: **end procedure***

Proposition 3. *Let $a \mapsto \mathbb{E}_m h(a, \omega): \mathbb{R}^{n+2} \to \mathbb{R}$ be a concave function that has a unique global maximizer a^*. Assume that a^* be a feasible action profile, i.e., $a^* \in \mathcal{A}$. Let α and β be two functions such that $\dot{\beta}(t) \leq e^{\alpha(t)}$. Consider the following Bregman learning algorithm*

$$\frac{d}{dt}\left[g_a\left(a(t) + e^{-\alpha(t)}\dot{a}(t)\right)\right] = e^{\alpha(t)+\beta(t)}\nabla_a\mathbb{E}_m h(a(t), \omega),$$
$$a(0) \in \mathbb{R}^{n+2}, \dot{a}(0) \in \mathbb{R}^{n+2}, \tag{13}$$

where $a(0)$ is the initial point of the algorithm and g is a strictly convex function on a. Let $a(t)$ be the solution to (13). Then the average regret within $[t_0, T], t_0 > 0$ is bounded above by

$$regret_T \leq \frac{c_0}{T - t_0}\int_{t_0}^{T} e^{-\beta(s)}ds, \tag{14}$$

where $c_0 := d_g(a^*, a(0)) + e^{-\alpha(0)}\dot{a}(0)) + e^{\beta(0)}\mathbb{E}_m[h(a^*, \omega) - h(a(0), \omega)] > 0$.

Proof. Let $W(a, \dot{a}, t, a^*) = d_g\left(a^*, a(t) + e^{-\alpha(t)}\dot{a}(t)\right) + e^{\beta(t)}\mathbb{E}_m[h(a^*, \omega) - h(a(t), \omega)]$. It is clear that W is positive. Moreover, $\frac{d}{dt}W(a(t), \dot{a}(t), t, a^*) \leq 0$ for $\dot{\beta} \leq e^{\alpha}$. Thus $W(a(t), \dot{a}(t), t, a^*) \leq W(a(0), \dot{a}(0), 0, a^*) = c_0$. By integration between $[t_0, T]$ it follows

$$\frac{1}{T - t_0}\int_{t_0}^{T}\mathbb{E}_m[h(a^*, \omega) - h(a(t), \omega)]dt \leq \frac{c_0}{T - t_0}\int_{t_0}^{T}e^{-\beta(s)}ds.$$

This completes the proof.

In particular, for $\beta(s) = -s + e^s$, one obtains an error bound to the minimum value as

$$\frac{c_0}{t}\int_0^t e^{-\beta(s)}ds = \frac{c_0}{t}\int_0^t e^s e^{-e^s}ds = \frac{c_0\left(\frac{1}{e} - e^{-e^t}\right)}{t},$$

and for $\beta(s) = s$, the regret bound becomes

$$\frac{c_0}{t}\int_0^t e^{-\beta(s)}ds = \frac{c_0(1 - e^{-t})}{t}.$$

Figure 1 illustrates the advantage of algorithm (13) compared with the gradient flow (10). It plots the regret bound $\frac{c_0}{T-t_0}\int_{t_0}^{T}e^{-\beta(s)}ds$ for $\beta = s$ and $d_g(a^*, a_0)\frac{\log\frac{T}{t_0}}{T-t_0}$ with an initial gap of $c_0 = 25$.

The advantage of algorithms (10) and (13) is that it is not required to compute the Hessian of $\mathbb{E}_m h(a, \omega)$ as it is the case in the Newton scheme. As a corollary of Proposition 2 the regret vanishes as T grows. Thus, it is a no-regret algorithm. However, Algorithm (10) may not be sufficiently fast. Algorithm (13) provides a higher order convergence rate by carefully design-ing (α, β). The average regret decays very quickly to zero [7]. However, it may generate an
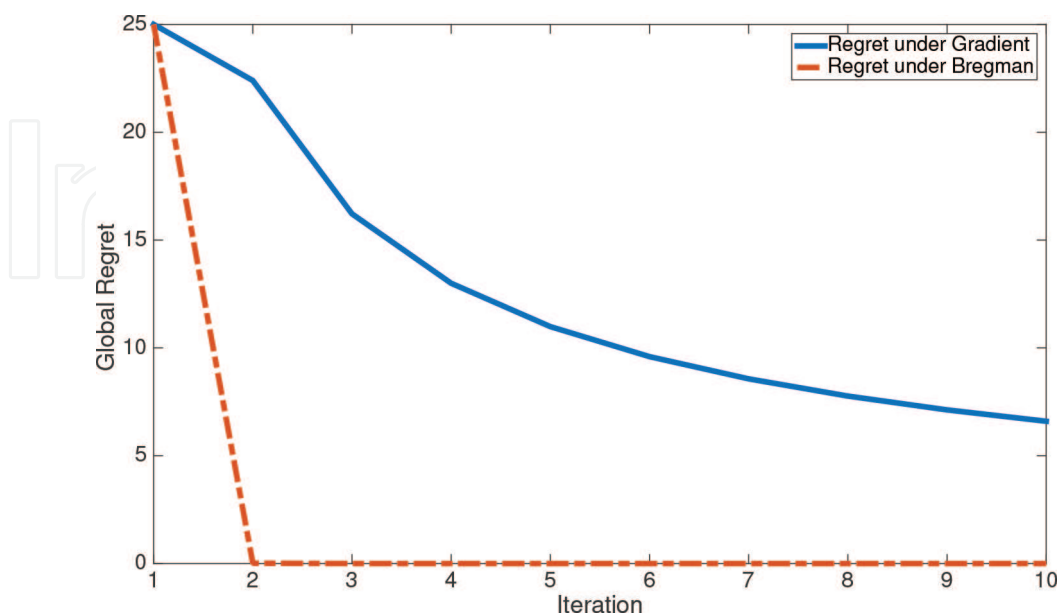
Figure 1. Global regret bound under Bregman vs. gradient. The initial gap is $c_0 = 25$.

oscillatory trajectory with a big magnitude. The next subsection presents risk-aware algorithms that reduce the oscillatory phase of the trajectory.

2.4.3. Risk-aware Bregman learning algorithm

In order to reduce the oscillatory phase, we introduce a risk-aware Bregman learning algorithm [7] which is a speed-up-and-average version of (13) called *mean dynamics* $\overline{m}$ of *a* given by

$$
\begin{aligned}
\overset{...}{\overline{m}} = &-\frac{3}{t}\overset{..}{\overline{m}} - (e^\alpha - \dot{\alpha})\left(\overset{..}{\overline{m}} + \frac{2}{t}\dot{\overline{m}}\right) \\
&+ \frac{e^{2\alpha+\beta}}{t} g_{\overline{m}\,\overline{m}}^{-1}\left(\overline{m} + [t + 2e^{-\alpha}]\dot{\overline{m}} + te^{-\alpha}\overset{..}{\overline{m}}\right)\mathbb{E}h_{\overline{m}}\left(t\dot{\overline{m}} + \overline{m}, \omega\right),
\end{aligned}
\tag{15}
$$

with starting vector $\overline{m}(0) = a(0), \dot{\overline{m}}(0), \overset{..}{\overline{m}}(0)$.

Algorithm 3. *The risk-aware Bregman learning pseudocode is as follows:*

1: **procedure** RISK-AWARE BREGMAN $(\overline{m}(0), \epsilon, T, g, \alpha, \beta, m, h)$ ▷ *The risk-aware Bregman learning starting from* $\overline{m}(0)$ *within* $[0, T]$

2:    $\overline{m} \leftarrow \overline{m}(0) = a(0), \dot{\overline{m}}(0), \overset{..}{\overline{m}}(0)$

3:    **while** *regret* $> \epsilon$ *and* $t \leq T$ **do** ▷ *We have the answer if regret is 0*

4:        *Compute* $\overline{m}(t)$ *solution of (15)*

5:        *Compute regret*

6:    **end while**

7:    **return** $\overline{m}(t), regret_t$ ▷ *get* $\overline{m}(t)$ *and the regret*

8: **end procedure**

Proposition 4. *The time-average trajectory of the learning algorithm (13) generates the mean dynamics (15).*

Proof. We use the average relation $\overline{m}(t) = \frac{1}{t}\int_0^t a(s)\,ds$ where *a* solves Eq. (13). From the definition of $\overline{m}$, and by Hopital's rule, $\overline{m}(0) = a(0)$. Moreover, $\overline{m}(t)$ and $a(t)$ share the following equations:

$$
a(t) = \overline{m}(t) + t\dot{\overline{m}}(t),
$$

$$
\dot{a}(t) = 2\dot{\overline{m}}(t) + t\overset{..}{\overline{m}}(t),
\tag{16}
$$

$$
\ddot{a}(t) = 3\overset{..}{\overline{m}}(t) + t\overset{...}{\overline{m}}(t).
$$

Substituting these values in Eq. (13) yields the mean dynamics (15). This completes the proof.

The risk-aware Bregman dynamics (15) generates a less oscillatory trajectory due to its averaging nature. The next result provides an accuracy bound for (15).

Proposition 5. *The risk-aware Bregman dynamics (15) satisfies*

$$0 \leq \mathbb{E}_m[h(a^*, \omega) - h(\overline{m}(t), \omega)] \leq \frac{c_0}{t} \int_0^t e^{-\beta(s)} ds.$$

Proof. Let $\overline{m}(t) = \frac{1}{t} \int_0^t a(s) ds$. Then, $\overline{m}(t) = \int_{\mathbb{R}} a(s) \left(\frac{1}{t} 1\!1_{[0,t]}(s) \right) ds$. Thus, $\overline{m}(t) = \mathbb{E}_{\mu(t)} a$ where $\mu(t)$ is the measure with density $d\mu(t)[s] = \frac{1}{t} 1\!1_{[0,t]}(ds)$. By convexity of $-\mathbb{E}_m h(a, \omega)$ we apply the Jensen's inequality:

$$\mathbb{E}_m h \left(\frac{1}{t} \int_0^t a(s) ds, \omega \right) = \mathbb{E}_m h(\overline{m}(t), \omega) = \mathbb{E}_m h \left(\mathbb{E}_{\mu(t)} a, \omega \right)$$

$$\geq \mathbb{E}_{\mu(t)} \mathbb{E}_m h(a, \omega) = \frac{1}{t} \int_0^t \mathbb{E}_m h(a(s), \omega) ds.$$

In view of (14) one has

$$0 \leq \mathbb{E}_m h(a^*, \omega) - \mathbb{E}_m h \left(\frac{1}{t} \int_0^t a(s) ds, \omega \right)$$

$$\leq \frac{1}{t} \int_0^t [\mathbb{E}_m h(a^*, \omega) - \mathbb{E}_m h(a(s), \omega)] ds$$

$$\leq c_0 \frac{1}{t} \int_0^t e^{-\beta(s)} ds,$$

$$0 \leq \mathbb{E}_m h(a^*, \omega) - \mathbb{E}_m h(\overline{m}(t), \omega) \leq \frac{c_0}{t} \int_0^t e^{-\beta(s)} ds.$$

This completes the proof.

Definition 5. (Convergence time). *Let $\delta > 0$ and $a(t)$ be the trajectory generated by Bregman algorithm starting from a_0 at time t_0. The convergence time to be within a ball $B(\mathbb{E}_m h(a^*, \omega), \delta)$ of radius $\delta > 0$ from the center $r(a^*)$ is given by*

$$T_\delta = \inf\{t \mid \mathbb{E}_m[h(a^*, \omega) - h(a(t), \omega)] \leq \delta, \ t > t_0\}.$$

Proposition 6. *Under the assumptions above, the error generated by the algorithm is at most (14) which means that it takes at most $T_\delta = \beta^{-1} \left[\log \frac{c_0}{\delta} \right]$ time units to the algorithm to be within a ball $B(r(a^*), \delta)$ of radius $\delta > 0$ from the center $\mathbb{E}_m h(a^*, \omega)$.*

*Proof.* The proof is immediate. For $\delta > 0$ the average regret bound of Proposition 5,

Convergence	Error bound	Time-to-reach T_δ
Triple exponential	$e^{-e^{e^t}}c_0$	$\log\left[\log\left(\log\frac{c_0}{\delta}\right)\right]$
$\alpha(t) = t + e^t, \ \beta(t) = e^{e^t}$		
Double exponential rate	$e^{-e^t}c_0$	$\log\left(\log\frac{c_0}{\delta}\right)$
$\alpha(t) = t, \ \beta(t) = e^t$		
Exponential rate	$e^{-t}c_0$	$\log\frac{c_0}{\delta}$
$\alpha(t) = 0, \ \beta(t) = t$		
Polynomial order k	$\frac{c_0}{t^k}$	$\frac{c_0^{1/k}}{\delta^{1/k}}$
$\alpha(t) = \log k - \log t, \ \beta(t) = k\log t$		

Table 1. Convergence rate under different set of functions.

$$regret_T \leq \frac{c_0}{T - t_0}\int_{t_0}^{T} e^{-\beta(s)}ds \leq \delta, \tag{17}$$

provides the announced convergence time bound. This completes the proof.

See **Table 1** for detailed parametric functions on the bound T_δ.
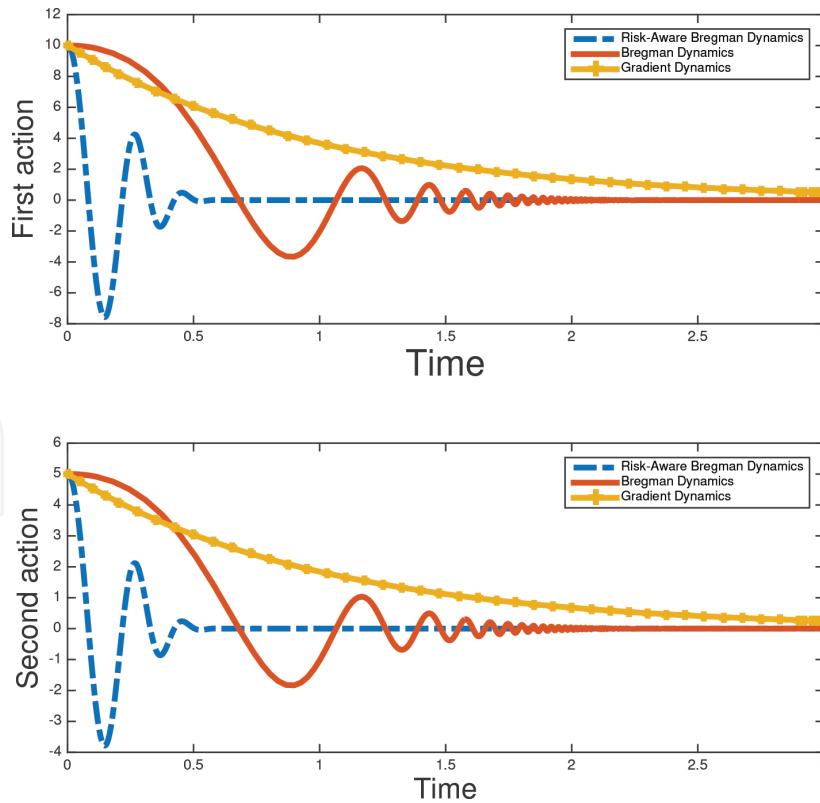
Figure 2. Gradient ascent vs. risk-aware Bregman dynamics for $r = -\left(1 + \sum_{k=1}^{2}\omega_k^2 a_k^2\right)$.

Example 4. *Let $f(y) = y \log y$ defined on $\mathbb{R}_+^*$. Then, $f(1) = 0$, and derivatives of f are $f'(y) = 1 + \log y$, $f''(y) = \frac{1}{y} > 0$. The Legendre-Fenchel transform of f is $f^*(\xi) = y^* = e^{\xi - 1}$. Let $(a_1, a_2) \mapsto g(a) = \|a\|_2^2$, and $(a_1, a_2, \omega) \mapsto r(a_1, a_2, \omega) = -\left(1 + \sum_{k=1}^2 \omega_{ik}^2 a_k^2\right)$. The coefficient ω distribution is unknown but a sampled empirical measure m is considered to be similar to uniform distribution in $(0, 1]$ with 10^4 samples. We illustrate the quick convergence rate of the algorithm in a basic example and plot in **Figure 2** the trajectories under standard gradient, Bregman dynamics and risk-aware Bregman dynamics (15). In particular, we observe that risk-aware Bregman dynamics (15) provides very quickly a satisfactory value. In this particular setup, we observe that the accuracy of the risk-aware Bregman algorithm (15) at $t = 0.5$ will need four times ($t = 2$) less than the standard Bregman algorithm to reach a similar level of error. It takes 40 times more ($t = 20$) than the gradient ascent to reach that level. Also, we observe that the risk-aware Bregman algorithm is less oscillatory and the amplitude decays very fast compared to the risk-neutral algorithm.*

3. Constrained distributionally robust optimization

In the constrained case i.e., when $\mathcal{A}$ is a strict subset of $\mathbb{R}^{n+2}$, algorithms (10) and (13) present some drawbacks: The trajectory $a(t)$ may not be feasible, i.e., $a(t) \notin \mathcal{A} \times \mathbb{R}_+ \times \mathbb{R}$ even when it starts in $\mathcal{A}$. In order to design feasible trajectories, projected gradient has been widely studied in the literature. However, a projection into $\mathcal{A}$ at each time t involves additional optimization problems and the computation of the projected gradient adds extra complexity to the algorithm. We restrict our attention to the following constraints:

$$\mathcal{A} = \left\{ a \in \mathbb{R}^n \mid a_l \in \left[\underline{a}_l, \overline{a}_l\right], \ l \in \{1, ..., n\}, \ \sum_{l=1}^n c_l a_l \leq b \right\}.$$

We impose the following feasibility condition: $\underline{a}_l < \overline{a}_l$, $l \in \{1, ..., n\}, c_l > 0$, $\sum_{l=1}^n c_l \underline{a}_l < b$. Under this setting, the constraint set $\mathcal{A}$ is non-empty, convex and compact.

We propose a method to compute a constrained solution that has a full support (whenever it exists). We do not use the projection operator. Indeed we transform the domain $\left[\underline{a}_l, \overline{a}_l\right] = \xi([0, 1])$ where $\xi(x_l) = \overline{a}_l x_l + \underline{a}_l (1 - x_l) = a_l$. ξ is a one-to-one mapping and

$$x_l = \xi^{-1}(a_l) = \frac{a_l - \underline{a}_l}{\overline{a}_l - \underline{a}_l} \in [0, 1].$$

$$\sum_{l=1}^n c_l (\overline{a}_l - \underline{a}_l) x_l \leq b - \sum_{l=1}^n c_l \underline{a}_l =: \hat{b}.$$

The algorithm (18)

$$\begin{cases} \dot{y} = \left[\nabla^2 g\right]^{-1} \nabla_a \mathbb{E}_m h(a, \omega) =: \hat{f}(a), \\ a_l := \bar{a}_l x_l + \underline{a}_l (1 - x_l), \\ x_l = \min\left(1, \dfrac{e^{y_l}}{\sum_{k=1}^{n} e^{y_k}} \dfrac{\hat{b}}{\left[c_l\left(\bar{a}_l - \underline{a}_l\right)\right]}\right), \\ l \in \{1, \dots, n\}, \end{cases} \tag{18}$$

generates a trajectory $a(t)$ that satisfies the constraint.

Algorithm 4. *The constrained learning pseudocode is as follows:*

1: **procedure** CONSTRAINED GRADIENT $(a(0), \epsilon, T, g, m, h)$ ▷ *The constrained learning algorithm starting from $a(0)$ within $[0, T]$*

2: $a \leftarrow a(0)$

3:   **while** *regret* $> \epsilon$ *and* $t \leq T$ **do** ▷ *We have the answer if regret is 0*

4:   *Compute $a(t)$ solution of (18)*

5:   *Compute regret*

6:   **end while**

7:   **return** $a(t), \text{regret}_t$ ▷ *get a(t) and the regret*

8: **end procedure**

Proposition 7. *If $\hat{b} \leq \min_l c_l\left(\bar{a}_l - \underline{a}_l\right)$ then Algorithm (18) reduces to*

$$\begin{cases} a_l := \bar{a}_l x_l + \underline{a}_l (1 - x_l), \\ \dot{x}_l = x_l \left[\left\langle e_l, \hat{f}(a)\right\rangle - 1\hat{b} \sum_l \left\langle e_l, \hat{f}(a)\right\rangle x_l \left[c_l\left(\bar{a}_l - \underline{a}_l\right)\right]\right], \\ l \in \{1, \dots, n\} \end{cases} \tag{19}$$

Proof. It suffices to check that for $\hat{b} \leq \min_l c_l\left(\bar{a}_l - \underline{a}_l\right)$, the vector z defined by $z_l = \dfrac{e^{y_l}}{\sum_{k=1}^{n} e^{y_k}}$ solves the replicator equation,

$$\dot{z}_l = z_l \left[\dot{y}_l - \langle z, \dot{y}\rangle\right].$$

Thus, $x_l = \dfrac{e^{y_l}}{\sum_{k=1}^{n} e^{y_k}} \dfrac{\hat{b}}{\left[c_l\left(\bar{a}_l - \underline{a}_l\right)\right]}$ solves $\dot{x}_l = x_l \left[\left\langle e_l, \hat{f}(a)\right\rangle - 1\hat{b} \sum_l \left\langle e_l, \hat{f}(a)\right\rangle x_l \left[c_l\left(\bar{a}_l - \underline{a}_l\right)\right]\right]$. This completes the proof.

Note that the dynamics of x in Eq. (19) is a constrained replicator dynamics [8] which is widely used in evolutionary game dynamics. This observation establishes a relationship between optimization and game dynamics and explains that the replicator dynamics is the gradient flow of the (expected payoff) under simplex constraint.

The next example illustrates a constrained distributionally robust optimization in wireless communication networks.

Example 5 (Wireless communication). *Consider a power allocation problem over n medium access channels. The signal-to-interference-plus-noise ratio (SINR) is*

$$SINR_l = \frac{\frac{a_l |\omega_{ll}|^2}{\left(d^2(s_r(l), s_t(l)) + \varepsilon^2\right)^{\frac{o}{2}}}}{N_0(s_r(l)) + I_l(s_r(l))},$$

where

- $N_0 > 0$ is the background noise.

- The interference on channel l is denoted $I_l \geq 0$. One typical model for I_l is

$I_l = \sum_{k \neq l} \frac{a_k |\omega_{kl}|^2}{\left(d^2(s_r(l), s_t(k)) + \varepsilon^2\right)^{\frac{o}{2}}}$.

- $\epsilon > 0$ is the height of the transmitter antenna.

- ω_{ll} is the channel state at l. The channel state is unknown. Its true distribution is also unknown.

- $s_r(l)$ is the location of the receiver of l

- $s_t(l)$ is the location of the transmitter of l

- $o \in \{2, 3, 4\}$ is the pathloss exponent.

- a_l is the power allocated to channel l. It is assumed to be between $\underline{a}_l \geq 0$ and $\overline{a}_l$ with $0 \leq \underline{a}_l < \overline{a}_l < +\infty$. Moreover, a total power budget constraint is imposed $\sum_{l=1}^{n} a_l \leq \overline{a}$ where $\overline{a} > \sum_{l=1}^{n} \underline{a}_l \geq 0$.

It is worth mentioning that the action constraint of the power allocation problem are similar to the ones analyzed in Section 3. The admissible action space is

$$\mathcal{A} := \left\{ a \in \mathbb{R}_+^n : \quad \underline{a}_l \leq a_l \leq \overline{a}_l, \sum_{l=1}^{n} a_l \leq \overline{a} \right\}.$$

Clearly, $\mathcal{A}$ is a non-empty convex compact set. The payoff function is the sum-rate $r(a, \omega) = \sum_{l=1}^{n} W_l \log(1 + SINR_l)$ where $W_l > 0$. The mapping $(a, \omega) \mapsto r(a, \omega)$ is continuously differentiable.

- Robust *optimization is too conservative: Part of the robust optimization problem [9, 7] consists of choosing the channel gain $|\omega_{ll}|^2 \in [0, \overline{\omega}_{ll}]$ were the bound $\overline{\omega}$ need to be carefully designed. However the worst case is achieved when the channel gain is zero: $\inf_{\omega \in \prod_l [0, \overline{\omega}_{ll}]} r(a, \omega) = 0$. Hence the robust performance is zero. This is too conservative as several realizations of the channel may give better performance than zero. Another way is to re-design the bounds $\underline{\omega}_{ll}$ and $\overline{\omega}_{ll}$. But if $\underline{\omega}_{ll} > 0$ it means that very low channel gains are not allowed, which may be too optimistic. Below we use the distributional robust optimization approach which eliminates this design issue.*

- Distributional *robust optimization: By means of the training sequence or channel estimation method, a certain (statistical) distribution m is derived. However m cannot be considered as the*

true distribution of the channel state due to estimation error. The true distribution of ω is unknown. Based on this observation, an uncertainty set $B_\rho(m)$ with radius $\rho \geq 0$ is constructed for alternative distribution candidates. Note that $\rho = 0$ means that $B_0(m) = \{m\}$. The distributional robust optimization problem is $\sup_a \inf_{\tilde{m} \in B_\rho(m)} \mathbb{E}_{\tilde{m}} r(a, \omega)$. In presence of interference, the function $r(a, \omega)$ is not necessarily concave in a. In absence of interference, the problem becomes concave.

4. Distributed optimization

This section presents distributed distributionally robust optimization problems over a direct graph. A large number of virtual agents can potentially choose a node (vertex) subject to constraint. The vector a represents the population state. Since a has n components, the graph has n vertices. The interactions between virtual agents are interpreted as possible connections of the graph. Let us suppose that the current interactions are represented by a directed graph $\mathcal{G} = (\mathcal{L}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{L}^2$ is the set of links representing the possible interaction among the proportion of agents, i.e., if $(l, k) \in \mathcal{E}$, then the component l of a can interact with the k−th component of a. In other words, $(l, k) \in \mathcal{E}$ means that virtual agents selecting the strategy $l \in \mathcal{L}$ could migrate to strategy $k \in \mathcal{L}$. Moreover, $\Lambda \in \{0, 1\}^{n \times n}$ is the adjacency matrix of the graph $\mathcal{G}$, and whose entries are $\lambda_{lk} = 1$, if $(l, k) \in \mathcal{E}$; and $\lambda_{lk} = 0$, otherwise.

Definition 6. *The distributionally robust fitness function is the marginal distributionally robust payoff function. If $a \mapsto \mathbb{E}_m h(a, \omega)$ is continuously differentiable, the distributionally robust fitness function is $\mathbb{E}_m \nabla_a h(a, \omega)$.*

Definition 7. *The virtual population state a is an equilibrium if $a \in \mathcal{A}$ and it solves the variational inequality*

$$\langle a - b, \mathbb{E}_m \nabla_a h(a, \omega) \geq 0, \quad \forall b \in \mathcal{A}.$$

Proposition 8. *Let the set of virtual population state $\mathcal{A}$ be non-empty convex compact and $b \mapsto \mathbb{E}_m \nabla h(b, \omega)$ be continuous. Then the following conditions are equivalent:*

- $\langle a - b, \mathbb{E}_m \nabla h(a, \omega) \geq 0, \quad \forall b \in \mathcal{A}.$

- *the action a satisfies $a = proj_{\mathcal{A}}[a + \eta \mathbb{E}_m \nabla h(a, \omega)]$*

Proof. Let a be a feasible action that solves the variational inequality:

$$\langle a - b, \mathbb{E}_m \nabla h(a, \omega) \geq 0, \quad \forall b \in \mathcal{A}.$$

Let $\eta > 0$. By multiplying both sides by η, we obtain

$$\langle a - b, \eta \mathbb{E}_m \nabla h(a, \omega) \geq 0, \quad \forall b \in \mathcal{A}.$$

We add the term $\langle a, b-a \rangle$ to both sides to obtain the following relationships:

$$
\begin{aligned}
&\langle a-b, \eta \mathbb{E}_m \nabla h(a, \omega) \rangle \geq 0 && \forall b \in \mathcal{A}, \\
&\Leftrightarrow \langle a-b, \eta \mathbb{E}_m \nabla h(a, \omega) \rangle + \langle a-b, -a \rangle \geq \langle a, b-a \rangle && \forall b \in \mathcal{A}, \\
&\Leftrightarrow \langle b-a, -[a + \eta \mathbb{E}_m \nabla h(a, \omega)] \rangle + \langle a-b, -a \rangle \geq 0 && \forall b \in \mathcal{A}, \\
&\Leftrightarrow \langle b-a, a - [a + \eta \mathbb{E}_m \nabla h(a, \omega)] \rangle \geq 0 && \forall b \in \mathcal{A},
\end{aligned}
\tag{20}
$$

Recall that the projection operator on a convex and closed set $\mathcal{A}$ is uniquely determined by

$$
z \in \mathbb{R}^n, z' = proj_{\mathcal{A}}[z] \Leftrightarrow \langle z' - z, b - z' \rangle \geq 0, \quad \forall b \in \mathcal{A}.
$$

Thus

$$
\begin{aligned}
&\langle b-a, a - [a + \eta \mathbb{E}_m \nabla h(a, \omega)] \rangle \geq 0, \quad \forall b \in \mathcal{A} \\
&\Leftrightarrow a = proj_{\mathcal{A}}[a + \eta \mathbb{E}_m \nabla h(a, \omega)].
\end{aligned}
\tag{21}
$$

This completes the proof.

As a consequence we can derive the following existence result.

Proposition 9. *Let the set of virtual population states $\mathcal{A}$ be a non-empty convex compact and the mapping $b \mapsto \mathbb{E}_m \nabla h(b, \omega)$ be continuous. Then, there exists at least one equilibrium in $\mathcal{A}$.*

Proof. A direct application of the Brouwer-Schauder's fixed-point theorem which states that if $\phi : \mathcal{A} \to \mathcal{A}$ is continuous and $\mathcal{A}$ non-empty convex compact then ϕ has at least one fixed-point in $\mathcal{A}$. Here we choose $\phi(a) = proj_{\mathcal{A}}[a + \eta \mathbb{E}_m \nabla h(a, \omega)]$. Clearly $\phi(\mathcal{A}) \subseteq \mathcal{A}$ and ϕ is continuous on $\mathcal{A}$ as the mapping $b \mapsto \mathbb{E}_m \nabla h(b, \omega)$ and the projection operator $b \mapsto proj_{\mathcal{A}}[b]$ are both continuous. Then the announced result follows. This completes the proof.

Note that we do not need sophisticated set-valued fixed-point theory to obtain this result.

Definition 8. *The virtual population state a is evolutionarily stable if $a \in \mathcal{A}$ and for any alternative deviant state $b \neq a$ there is an invasion barrier $\epsilon_b > 0$ such that*

$$
\langle a-b, \mathbb{E}_m \nabla h(a + \epsilon(b-a), \omega) \rangle > 0, \quad \forall \epsilon \in (0, \epsilon_b).
$$

The function $\varrho : \mathcal{A} \times \mathbb{R}^n \times \mathbb{R}_+^{n \times n} \to \mathbb{R}^{n \times n}$ is the revision protocol, which describes how virtual agents are making decisions. The revision protocol ϱ takes a population state a, the corresponding fitness $\nabla \mathbb{E}_m h$, the adjacency matrix Λ and returns a matrix. Therefore, let $\varrho_{lk}(a, h, \Lambda)$ be the switching rate from the lth to k^{th} component. Then, the virtual agents selecting the strategy $l \in \mathcal{L}$ have incentives to migrate to the strategy $l \in \mathcal{L}$ only if $\varrho_{lk}(a, h, \Lambda) > 0$, and it is also possible to design switch rates depending on the topology describing the migration constraints, i.e., $\lambda_{lk} = 0 \Rightarrow \varrho_{lk}(a, h, \Lambda) = 0$. The distributed distributionally robust optimization consists to perform the optimization problem above over the distributed network that is

subject to communication restriction. We construct a distributed distributionally robust game dynamics to perform such a task. The distributed distributionally robust evolutionary game dynamics emerge from the combination of the (robust) fitness h and the constrained switching rates ϱ. The evolution of the portion a_l is given by the distributed distributional robust mean dynamics

$$\dot{a}_l = \sum_{k \in \mathcal{L}} a_k \varrho_{kl}(a, h, \Lambda) - a_l \sum_{k \in \mathcal{L}} \varrho_{lk}(a, h, \Lambda), l \in \mathcal{L}, \tag{22}$$

Since the distributionally robust function h is obtained after the transformation from payoff function r by means of triality theory, the dynamics (22) is seeking for distributed distributionally robust solution.

Algorithm 5. *The distributed distributional robust mean dynamics pseudocode is as follows:*

*1: **procedure** POPULATION-INSPIRED ALGORITHM $(a(0), \epsilon, T, \varrho, g, m, h, \Lambda) \triangleright$ The population-inspired learning starting from $a(0)$ within $[0, T]$*

2: $a \leftarrow a(0)$

*3:   **while** regret $> \epsilon$ and $t \leq T$ **do** $\triangleright$ We have the answer if regret is 0*

4:   Compute $a(t)$ solution of (22)

5:   Compute $regret_t$

*6:   **end while***

*7:   **return** $a(t), regret_t \triangleright$ get $a(t)$ and the regret*

*8: **end procedure***

The next example establishes evolutionarily stable state, equilibria and rest-point of the dynamics (22) by designing ϱ.

Example 6. *Let us consider a power system that is composed of 10 generators, i.e., let $\mathcal{L} = \{1, \dots, 10\}$. Let $a_l \in \mathbb{R}_+$ be the power generated by the generator $l \in \mathcal{L}$. Each power generation should satisfy the physical and/or operation constraints $a_l \in [\underline{a}_l, \overline{a}_l]$, for all $l \in \mathcal{L}$. It is desired to satisfy the power demand given by $d \in \mathbb{R}$, i.e., it is necessary to guarantee that $\sum_{l \in \mathcal{L}} a_l = d$, i.e., the supply meets the demand. The objective is to minimize the generation quadratic costs for all the generators, i.e.,*

$$\text{Maximize} \quad r(a, \omega) = \sum_{l \in \mathcal{L}} r_l(a_l) = -\sum_{l \in \mathcal{L}} \left(c_{0l} + c_{1l} a_l + c_{2l} a_l^2 \right),$$

$$\text{s.t.} \quad \sum_{l \in \mathcal{L}} a_l = d, \quad \underline{a}_l \leq a_l \leq \overline{a}_l, \ l \in \mathcal{L},$$

where $r : \mathbb{R}^n \to \mathbb{R}$ is concave, and the parameters are possibly uncertain and selected as $c_{0l} = 25 + 6l$, $c_{1l} = 15 + 4l + \omega_{1l}$, $c_{2l} = 5 + l + \omega_{2l}$, and $d = 20 + \omega_{3l}$. Therefore, the fitness

functions for the corresponding full potential game are given by $f_l(a) = -2a_l c_{2l} - c_{1l}$, for all $l \in \mathcal{L}$, and action space is given by

$$\mathcal{A} = \left\{ a \in \mathbb{R}_+^n : \sum_{l \in \mathcal{L}} a_l = d, \ a_l \in [\underline{a}_l, \overline{a}_l] \right\}.$$

The distributed revision protocol is set to

$$\varrho_{lk}(a, h, \Lambda) = \frac{\lambda_{lk}}{a_l} \max(0, \overline{a}_k - a_k) \max(0, a_l - \underline{a}_l) \max(0, \mathbb{E}_m(h_k - h_l)),$$

for $a_l \neq 0$. We evaluate four different scenarios, i.e.,

1. $\underline{a} = 0_n$ and $\overline{a} = d \mathbb{1}_n$,

2. $\underline{a}_l = 0$, for all $l \in \mathcal{L} \ \{9, 10\}$, $\underline{a}_9 = 1.1$, and $\underline{a}_{10} = 1$; and $\overline{a}_l = d$, for all $l \in \mathcal{L} \ \{1, 2\}$, $\overline{a}_1 = 3$, and $\overline{a}_2 = 2.5$,

3. *Case 1 constraints and with interaction restricted to the cycle graph $\mathcal{G} = (\mathcal{L}, \mathcal{E})$ with set of links $\mathcal{E} = \left\{ \cup_{l \in \mathcal{L} \ \{n\}} (l, l+1) \right\} \cup \{(n, 1)\}$,*

4. *Case 2 constraints and with interaction restricted as in Case 3.*

Figure 3 presents the evolution of the generated power, the fitness functions corresponding to the marginal costs and the total cost. For the first scenario, the evolutionary game dynamics converge to a standard evolutionarily stable state in which $\hat{f}(a^\star) = c \mathbb{1}_n$. In contrast, for the second scenario, the dynamics converge to a constrained evolutionarily stable state.

4.1. Extension to multiple decision-makers

Consider a constrained game $\mathcal{G}$ in strategic-form given by

- $\mathcal{P} = \{1, \dots, P\}$ is the set of players. The cardinality of $\mathcal{P}$ is $P \geq 2$.

- Player p has a decision space $A_p \subset \mathbb{R}^{n_p}$, $n_p \geq 1$. Players are coupled through their actions and their payoffs. The set of all feasible action profiles is $\mathcal{A} \subset \mathbb{R}^n$, with $n = \sum_{p \in \mathcal{P}} n_p$. Player p can choose an action a_p in the set $\mathcal{A}_p(a_{-p}) = \{a_p \in A_p : (a_p, a_{-p}) \in \mathcal{A}\}$.

- Player p has a payoff function $r_p : \mathcal{A} \to \mathbb{R}$.

We restrict our attention to the following constraints:

$$A_p = \left\{ a_p \in \mathbb{R}^{n_p} \ | \ a_{pl} \in \left[\underline{a}_{pl}, \overline{a}_{pl} \right], \ l \in \{1, \dots, n_p\}, \ \sum_{l=1}^{n_p} c_{pl} a_{pl} \leq b_p \right\}$$

The coupled constraint is

$$\mathcal{A} = \left\{ a \in \prod_p A_p, \ \sum_{p \in \mathcal{P}} \langle \overline{c}_p, a_p \rangle \leq \overline{b} \right\}.$$
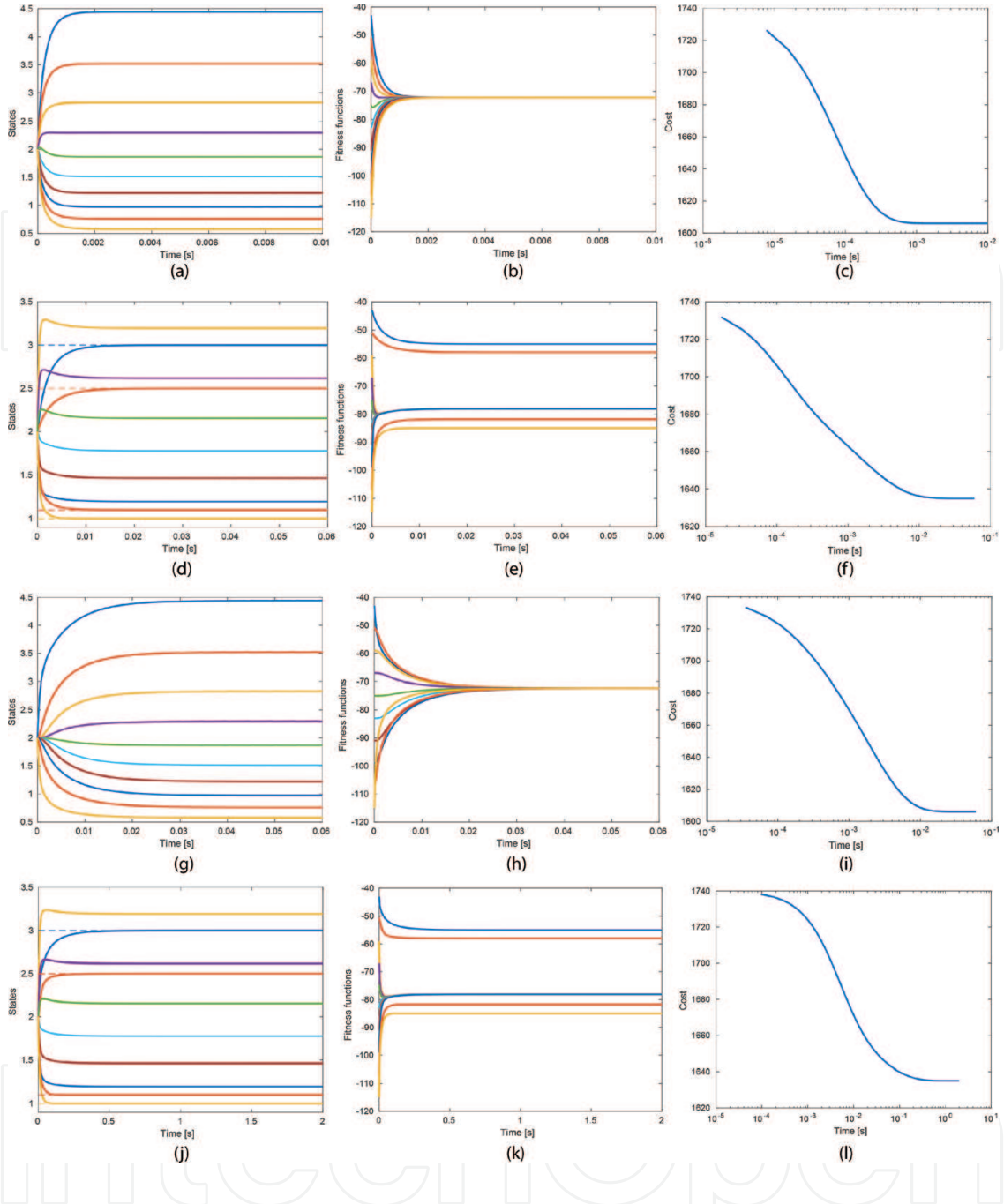
Figure 3. Economic power dispatch. Evolution of the population states (generated power), fitness functions $\hat{f}(a) = \nabla \mathbb{E}h(a, \omega)$, and the costs $-\mathbb{E}r(a, \omega)$. Figures (a)-(c) for case 1, (d)-(f) for case 2, (g)-(i) for case 3, and (j)-(l) for case 4.

Feasibility condition: If $\underline{a}_{pl} < \bar{a}_{pl}$, $l \in \{1, \ldots, n_p\}$, $c_{pl} > 0$, $\sum_{l=1}^{n_p} c_{pl}\underline{a}_{pl} < b_p$, $\bar{c}_p \in \mathbb{R}_{>0}^{n_p}$ and $\sum_{p \in \mathcal{P}} \langle \bar{c}_p, \underline{a}_p \rangle < \bar{b}$, the constraint set $\mathcal{A}$ is non-empty, convex and compact.

We propose a method to compute a constrained equilibrium that has a full support (whenever it exists). We do not use the projection operator. Indeed we transform the domain $\left[\underline{a}_{pl}, \bar{a}_{pl}\right] = \xi([0, 1])$ where $\xi(x_{pl}) = \bar{a}_{pl}x_{pl} + \underline{a}_{pl}(1 - x_{pl}) = a_{pl}$. ξ is a one-to-one mapping and

$$x_{pl} = \xi^{-1}\left(a_{pl}\right) = \frac{a_{pl} - \underline{a}_{pl}}{\overline{a}_{pl} - \underline{a}_{pl}} \in [0, 1].$$

$$\sum_{l=1}^{n_p} c_{pl}\left(\overline{a}_{pl} - \underline{a}_{pl}\right) x_{pl} \leq b_p - \sum_{l=1}^{n_p} c_{pl}\underline{a}_{pl} =: \hat{b}_p.$$

The learning algorithm (23) is

$$\begin{cases} \dot{y}_p = \left[\nabla_p^2 g\right]^{-1} \nabla_{a_p} r_p(a, \omega), \\ a_{pl} := \overline{a}_{pl}x_{pl} + \underline{a}_{pl}\left(1 - x_{pl}\right), \\ x_{pl} = \min\left(1, \frac{e^{y_{pl}}}{\sum_{k=1}^{n_p} e^{y_{pk}}} \frac{\hat{b}_p}{\left[c_{pl}\left(\overline{a}_{pl} - \underline{a}_{pl}\right)\right]}\right), \\ l \in \{1, \ldots, n_p\}, \end{cases} \tag{23}$$

generates a trajectory $a_p(t) = \left(a_{pl}(t)\right)_l$ that satisfies the constraint of player p at any time t.

5. Notes

The work in [10] provides a nice intuitive introduction to robust optimization emphasizing the parallel with static optimization. Another nice treatment [11], focusing on robust empirical risk minimization problem, is designed to give calibrated confidence intervals on performance and provide optimal tradeoffs between bias and variance [12, 13]. f-divergence based performance evaluations are conducted in [11, 14, 15]. The connection between risk-sensitivity measures such as the exponentiated payoff and distributionally robustness can be found in [16]. Distributionally robust optimization and learning are extended to multiple strategic decision-making problems i.e., distributionally robust games in [17, 18].

Acknowledgements

Author details

Jian Gao, Yida Xu, Julian Barreiro-Gomez, Massa Ndong, Michalis Smyrnakis and Hamidou Tembine*

*Address all correspondence to: tembine@ieee.org

Learning and Game Theory Laboratory, New York University, Abu Dhabi, United Arab Emirates

References

[1] Sion M. On general minimax theorems. Pacific Journal of Mathematics. 1958;**8**(1):171-176

[2] Bach FR. Duality between subgradient and conditional gradient methods. SIAM Journal on Optimization. 2015;**1**(25):115-129

[3] Kim D, Fessler JA. Optimized first-order methods for smooth convex minimization. Mathematical Programming. 2016;**159**(1):81-107

[4] Nesterov Y. Accelerating the cubic regularization of newton's method on convex problems. Mathematical Programming. 2008;**112**(1):159-181

[5] Nesterov Y. Primal-dual subgradient methods for convex problems. Mathematical Programming. 2009;**120**(1):221-259

[6] Larry A. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of Mathematics. Tokyo Japan: International Academic Printing Co., Ltd.; 1966;**16**(1):1-3

[7] Tembine H. Distributed Strategic Learning for Wireless Engineers. Boca Raton, FL, USA: CRC Press, Inc.; 2012. p. 496

[8] Taylor PD, Jonker L. Evolutionarily stable strategies and game dynamics. Mathematical Biosciences. 1978;**40**:145-156

[9] Ben-Tal A, Ghaoui LE, Nemirovski A. Robust Optimization. Princeton Series in Applied Mathematics. Princeton University Press; August 30, 2009. 576 p. ISBN-10: 0691143684, ISBN-13: 978-0691143682

[10] Esfahani PM, Kuhn D. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. In: Mathematical Programming. Jul 2017

[11] Duchi JC, Namkoong H. Stochastic gradient methods for distributionally robust optimization with f-divergences. Advances in Neural Information Processing Systems. 2015; 2208-2216

[12] Ben-Tal A, den Hertog D, Waegenaere AD, Melenberg B, Rennen G. Robust solutions of optimization problems affected by uncertain probabilities. Management Science. Catonsville, USA: Informs PubsOnLine; 2013;**59**(2):341-357

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

[14] Ahmadi-Javid A. An information-theoretic approach to constructing coherent risk measures. IEEE International Symposium on Information Theory Proceedings. 2011;2125-2127

[15] Ahmadi-Javid A. Entropic value-at-risk: A new coherent risk measure. Journal of Optimization Theory and Applications. 2012;**155**(3):1105-1123

[16] Föllmer H, Knispel T. Entropic risk measures: Coherence vs. convexity, model ambiguity, and robust large deviations. Stochastics Dynamics. 2011;**11**:333-351

[17] Bauso D, Gao J, Tembine H. Distributionally robust games: F-divergence and learning. ValueTools, International Conference on Performance Evaluation Methodologies and Tools, Venice, Italy; December 5-7, 2017

[18] Tembine H. Dynamic robust games in mimo systems. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). Aug 2011;**41**(4):990-1002