2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Online data assimilation in distributionally robust optimization*

D. Li[1] and S. Martínez[1]

Abstract— **This paper considers a class of real-time decision making problems to minimize the expected value of a function that depends on a random variable ξ under an unknown distribution $\mathbb{P}$. In this process, samples of ξ are collected sequentially in real time, and the decisions are made, using the real-time data, to guarantee out-of-sample performance. We approach this problem in a distributionally robust optimization framework and propose a novel ONLINE DATA ASSIMILATION ALGORITHM for this purpose. This algorithm guarantees the out-of-sample performance in high probability, and gradually improves the quality of the data-driven decisions by incorporating the streaming data. We show that the ONLINE DATA ASSIMILATION ALGORITHM guarantees convergence under the streaming data, and a criteria for termination of the algorithm after certain number of data has been collected.**

I. INTRODUCTION

Online data assimilation is of benefit in many applications that require real-time decision making under uncertainty, such as optimal target tracking, sequential planning problems, and robust quality control. In these problems, the uncertainty is often represented by a multivariate random variable that has an unknown distribution. Among available methods, distributionally robust optimization (DRO) has attracted attention due to its capability to handle data with unknown distributions while providing out-of-sample performance guarantees with limited uncertainty samples. To quantify the uncertainty and make decisions that guarantee the performance reliably, one often needs to gather a large number of samples in advance. Such requirement, however, is hard to achieve under scenarios where acquiring samples is expensive, or when real-time decisions must be made. Further, when the data is collected over time, it remains unclear what the best the procedure is to assimilate the data in an ongoing optimization process. Motivated by this, this work studies how to incorporate finitely streaming data into a DRO problem, while guaranteeing out-of-sample performance via the generation of time-varying certificates.

Literature Review: Optimization under uncertainty is a popular research area, and as such available methods include stochastic optimization [1] and robust optimization [2]. Recently, data-driven distributionally robust optimization has regained popularity thanks to its out-of-sample performance guarantees, see e.g. [3]–[6] and references therein. In this setup, one defines a set of distributions or *ambiguity set*, which contains the true distribution of the data-generating system with high probability. Then, the out-of-sample performance of the data-driven solution is obtained as the worst-case optimization over the ambiguity set. An attractive way of designing these sets is to consider a ball in the space of probability distributions centered at a reference or most-likely distribution constructed from the available data. In the space of distributions, the popular distance metric is the Prokhorov metric, ϕ-divergence and the Wasserstein distance [3], [5]. Here, following the paper [3], which proposes a distributed optimization algorithm for multi-agent settings, we use the Wasserstein distance as it leads to a tractable reformulation of DRO problems. However, available algorithms in [3], [5] do not consider the update of the data-driven solution over time, which serves as the focus of this work. In terms of the algorithm design, our work connects to various convex optimization methods [7] such as the Frank-Wolfe (FW) Algorithm (e.g., conditional gradient algorithm), the Subgradient Algorithm, and their variants, see e.g. [8], [9] and references therein. Our emphasis on the convergence of the data-driven solution obtained through a sequence of optimization problems contrasts with typical optimization algorithms developed for single (non-updated) problems.

Statement of Contributions: Our starting point is a distributionally robust optimization problem formulation setting of [3], [5], where we further consider that the limited realizations of the multivariate random variable in the problem are revealed and collected sequentially over time. As the probability distribution of the random variable is unknown, we aim to find and update a real-time data-driven solution based on streaming data. To guarantee the performance of the data-driven solution with certain reliability, we follow a DRO approach to solve a worst-case optimization problem that considers all the probability distributions in ambiguity sets given as a neighborhood of the empirical distribution under the Wasserstein metric. Our first contribution is the generation of such performance guarantee for any real-time data-driven solution. We achieve this by first finding an equivalent convex optimization problem over a simplex, and then specializing the algorithm for efficiently generating a performance certificate of the data-driven solution with a certain reliability requirement. Based on the fact that the performance guarantee of data-driven solution with high probability, our second contribution is the design of a scheme to find an optimal data-driven solution with the best performance guarantee under the same reliability. As new data is revealed and collected sequentially, we specialize the proposed scheme to assimilate the streaming data. We show that the resulting ONLINE DATA ASSIMILATION ALGORITHM is provably correct in the sense that the reliability of the out-of-sample performance guarantee for the generated data-driven solution converges to 1 as the number of data samples grows to infinity, and the data-driven solution with certain

*This research was developed with funding from the DARPA (Lagrange) contract N660011824027. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

[1]D. Li and S. Martínez are with the Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92092, USA lidan@ucsd.edu; soniamd@ucsd.edu

performance guarantee is available any time as soon as the algorithm finish generating the initial certificate. A convergence analysis of the proposed algorithm is given, under a user-defined optimality tolerance. We finally illustrate the performance of the proposed algorithm in simulation.

II. Preliminaries

Notations: Let $\mathbb{R}^m$, $\mathbb{R}^m_{\geq 0}$ and $\mathbb{R}^{m\times d}$ denote respectively the m-dimensional Euclidean space, the m-dimensional nonnegative orthant, and the space of $m \times d$ matrices, respectively. We use the shorthand notations $\mathbf{0}_m$ for the column vector $(0,\cdots,0)^\top \in \mathbb{R}^m$, $\mathbf{1}_m$ for the column vector $(1,\cdots,1)^\top \in \mathbb{R}^m$, and $\mathbf{I}_m \in \mathbb{R}^{m\times m}$ for the identity matrix. We let $\mathbf{x} \in \mathbb{R}^m$ denote a column vector with the dimension m and $\mathbf{x}^\top$ represents its transpose. We say a vector $\mathbf{x} \geq 0$, if all its the entries are nonnegative. We use subscripts to index vectors and superscripts to indicate the component of vector, i.e., $\mathbf{x}_k \in \mathbb{R}^m$ for $k \in \{1,2,\ldots,n\}$ and $\mathbf{x}_k := (\mathbf{x}_k^1,\ldots,\mathbf{x}_k^m)^\top$. We use $\mathbf{x}^{i:j}$ to denote the column vector $(\mathbf{x}^i,\mathbf{x}^{i+1},\ldots,\mathbf{x}^j)^\top \in \mathbb{R}^{j-i+1}$ and $(\mathbf{x};\mathbf{y}) \in \mathbb{R}^{m+d}$ indicates the concatenated column vector from $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^d$. An 1-norm of the vector $\mathbf{x} \in \mathbb{R}^m$ is denoted by $\|\mathbf{x}\|$. For matrices $A \in \mathbb{R}^{m\times d}$ and $B \in \mathbb{R}^{p\times q}$, we let $A \oplus B$ denote their direct sum. The shorthand notation $\oplus_{i=1}^m A_i$ represents $A_1 \oplus \cdots \oplus A_m$. Given a set of points I in $\mathbb{R}^m$, we let $\mathrm{conv}(I)$ indicate its convex hull. The gradient of a real-valued function $f : \mathbb{R}^m \to \mathbb{R}$ is written as $\nabla_{\mathbf{x}} f(\mathbf{x})$. The i^{th} component of the gradient vector is denoted by $\nabla_i f(\mathbf{x})$. We call the function f *proper* on $\mathbb{R}^m$ if $f(\mathbf{x}) < +\infty$ for at least one point $\mathbf{x} \in \mathbb{R}^m$ and $f(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in \mathbb{R}^m$. We use $\mathrm{dom}\, f$ to denote the effective domain of the proper function f, i.e., $\mathrm{dom}\, f := \{\mathbf{x} \in \mathbb{R}^m \mid f(\mathbf{x}) < +\infty\}$. We say a function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is *convex-concave* on $\mathcal{X} \times \mathcal{Y}$ if, for any point $(\tilde{\mathbf{x}},\tilde{\mathbf{y}}) \in \mathcal{X} \times \mathcal{Y}$, $\mathbf{x} \mapsto F(\mathbf{x},\tilde{\mathbf{y}})$ is convex and $\mathbf{y} \mapsto F(\tilde{\mathbf{x}},\mathbf{y})$ is concave.

Numerical Optimization Methods: There are mainly two types of Numerical Optimization methods that serve as the main ingredients of our Online Data Assimilation Algorithm. One type is given by Frank-Wolfe Algorithm (FWA) variants and the other is the Subgradient Algorithm. For the Subgradient Algorithm please refer to [10], [11].

The Frank-Wolfe Algorithm over a unit simplex. To solve convex programs over a unit simplex, here we introduce the FWA Algorithm following [8], [9]. We define the m-dimensional unit simplex as $\Delta_m := \{\lambda \in \mathbb{R}^m \mid \mathbf{1}_m{}^\top \lambda = 1,\ \lambda \geq 0\}$. Let Λ_m be the set of all extreme points for the simplex Δ_m. Consider the minimization of a convex function $f(\mathbf{x})$ over Δ_m; we refer to this problem by $(*)$ and denote by $\mathbf{x}^\star$ an optimizer of $(*)$. We refer to a $\mathbf{x}^\epsilon$ as an ϵ-optimal solution for $(*)$, if $\mathbf{x}^\epsilon \in \Delta_m$ and $f(\mathbf{x}^\epsilon) - f(\mathbf{x}^\star) \leq \epsilon$. We define a FW search point $\mathbf{s}^{(k)}$ for the current iteration k at the feasible point $\mathbf{x}^{(k)}$, if $\mathbf{s}^{(k)}$ is an extreme point such that $\mathbf{s}^{(k)} \in \mathrm{argmin}_{\mathbf{x}\in\Delta_m} \nabla f(\mathbf{x}^{(k)})^\top(\mathbf{x} - \mathbf{x}^{(k)})$. With this search point we define the FW direction at $\mathbf{x}^{(k)}$ by $d_{\text{FW}}^{(k)} := \mathbf{s}^{(k)} - \mathbf{x}^{(k)}$. The classical Frank-Wolfe Algorithm solves the problem $(*)$ to ϵ-optimality by iteratively finding a FW direction and then solving a line search problem over this direction until an ϵ-optimal solution $\mathbf{x}^{(k)}$ is found, certified by $\eta^{(k)} := -\nabla f(\mathbf{x}^{(k)}) \cdot d_{\text{FW}}^{(k)} \leq \epsilon$. Away-step Frank-Wolfe (AFW) Algorithm is an extension of the FWA we used in the following sections, and a linear convergence rate property of the AFW Algorithm is stated in the online version of this paper [12] for completeness.

III. Problem Description

Consider a decision-making problem given by

$$\inf_{\mathbf{x}\in\mathbb{R}^d} \mathbb{E}_{\mathbb{P}}[f(\mathbf{x},\xi)], \tag{P}$$

where the decision variable $\mathbf{x}$ on $\mathbb{R}^d$ is to be determined, the random variable $\xi : \Omega \to \mathbb{R}^m$ is induced by the probability space $(\Omega,\mathcal{F},\mathbb{P})$, and the expectation of f is taken w.r.t. the unknown distribution $\mathbb{P} \in \mathcal{M}(\mathcal{Z})$. It is not possible to evaluate the objective of (P) under $\mathbf{x}$ because $\mathbb{P}$ is unknown.

This section sets up the framework of an efficient Online Data Assimilation Algorithm that adapts the decision-making process by using streaming data, i.e., *independent and identically distributed* (iid) realizations of the random variable ξ. Then, we adapt the distributionally robust optimization approach following [3], [5], to complete the framework. We omit all proofs in the paper for simplicity and we just report an outline of the main ideas of the paper. Please see the online version [12] for more details.

Let $\{\hat{\mathbf{x}}^{(r)}\}_{r=1}^\infty$ be a sequence of decisions where for each iteration r the decision $\hat{\mathbf{x}}^{(r)}$ is feasible for (P). In our Online Data Assimilation Algorithm we generate $\{\hat{\mathbf{x}}^{(r)}\}_{r=1}^\infty$ while sequentially collecting iid realizations of the random variable ξ under $\mathbb{P}$, denoted by $\hat{\xi}_n$, $n = 1,2,\ldots$. This defines a sequence of streaming data sets, $\hat{\Xi}_n \subseteq \hat{\Xi}_{n+1}$, for each n. W.l.o.g. we assume that each data set $\hat{\Xi}_{n+1}$ consists of just one more new data point, i.e., $\hat{\Xi}_{n+1} = \hat{\Xi}_n \cup \{\hat{\xi}_{n+1}\}$ and $\hat{\Xi}_1 = \{\hat{\xi}_1\}$. The time between updates of $\hat{\Xi}_n$ and $\hat{\Xi}_{n+1}$ corresponds to certain time period, which we refer to as the n^{th} *time period*. The decision sequence obtained during this period is a subsequence of $\{\hat{\mathbf{x}}^{(r)}\}_{r=1}^\infty$, labeled by $\{\hat{\mathbf{x}}^{(r)}\}_{r=r_n}^{r_{n+1}}$. The objective of our algorithm is to make real-time decisions for (P) that have a potentially low objective value, while adapting the information from the current data set $\hat{\Xi}_n$.

To quantify the quality of the decisions $\{\hat{\mathbf{x}}^{(r)}\}_{r=1}^\infty$, we introduce the following terms. For each r and the n^{th} time period we call the decision $\hat{\mathbf{x}}^{(r)} \in \mathbb{R}^d$ *a proper data-driven solution* of (P), if $\hat{\mathbf{x}}^{(r)}$ is feasible and its *out-of-sample performance*, defined by $\mathbb{E}_{\mathbb{P}}[f(\hat{\mathbf{x}}^{(r)},\xi)]$, satisfies the following *performance guarantee*:

$$\mathbb{P}^n(\mathbb{E}_{\mathbb{P}}[f(\hat{\mathbf{x}}^{(r)},\xi)] \leq \hat{J}_n(\hat{\mathbf{x}}^{(r)})) \geq 1 - \beta_n, \tag{1}$$

where the *certificate* $\hat{J}_n$ is a function of $\hat{\mathbf{x}}^{(r)}$ that indicates the goodness of the performance under the data set $\hat{\Xi}_n$. The *reliability* $(1-\beta_n) \in (0,1) \subset \mathbb{R}$ governs the choice of the solution $\hat{\mathbf{x}}^{(r)}$ and the resulting certificate $\hat{J}_n(\hat{\mathbf{x}}^{(r)})$. Finding an approximate certificate is much easier than finding an exact certificate $\hat{J}_n$ in practice. Based on this, we call a solution $\hat{\mathbf{x}}^{(r)}$ ϵ_1-*proper*, if it satisfies (1) with a approximate certificate, $\hat{J}_n^{\epsilon_1}$, such that $\hat{J}_n(\hat{\mathbf{x}}^{(r)}) - \hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)}) \leq \epsilon_1$. The certificates $\hat{J}_n(\hat{\mathbf{x}}^{(r)})$ and $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$, which depend on $\hat{\mathbf{x}}^{(r)}$ and the data set $\hat{\Xi}_n$, provide an upper bound to the optimal

value of (P) with high confidence $(1-\beta_n)$ and are to be constructed carefully.

In each time period n, given a reliability level $1-\beta_n$, our goal is to approach to an ϵ_1-proper data-driven solution with a low certificate. Motivated by this we call any proper data-driven solution ϵ_2-*optimal*, labeled as $\hat{\mathbf{x}}_n^{\epsilon_2}$, if $\hat{J}_n(\hat{\mathbf{x}}_n^{\epsilon_2}) - \hat{J}_n(\mathbf{x}) \leq \epsilon_2$ for all $\mathbf{x} \in \mathbb{R}^d$. Then, for any ϵ_2-optimal and ϵ_1-proper data-driven solution $\hat{\mathbf{x}}_n^{\epsilon_2}$ with certificate $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}_n^{\epsilon_2})$ and $\epsilon_1 \ll \epsilon_2$, we have the following performance guarantee:

$$\mathbb{P}^n(\mathbb{E}_{\mathbb{P}}[f(\hat{\mathbf{x}}_n^{\epsilon_2}, \xi)] \leq \hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}_n^{\epsilon_2}) + \epsilon_1) \geq 1 - \beta_n. \tag{2}$$

We describe now the procedure of the ONLINE DATA ASSIMILATION ALGORITHM to solve (P). Given tolerance parameters ϵ_1 and ϵ_2, a sequence of data sets $\{\hat{\Xi}_n\}_{n=1}^N$ and strictly decreasing confidence levels $\{\beta_n\}_{n=1}^N$ with $N \to \infty$ such that $\sum_{n=1}^{\infty} \beta_n < \infty$, the algorithm aims to find a sequence of ϵ_2-optimal and ϵ_1-proper data-driven solutions, $\{\hat{\mathbf{x}}_n^{\epsilon_2}\}_{n=1}^N$, associated with the sequence of the certificates $\{\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}_n^{\epsilon_2})\}_{n=1}^N$ so that the performance guarantee (2) holds for all n. Additionally, as the data streams to infinity, i.e., $n \to \infty$ with $N = \infty$, there exists a large enough n_0 such that the algorithm terminates after processing the data set $\hat{\Xi}_{n_0}$. The algorithm returns a final data-driven solution $\hat{\mathbf{x}}_{n_0}^{\epsilon_2}$ such that the performance holds almost surely, i.e., $\mathbb{P}^{\infty}(\mathbb{E}_{\mathbb{P}}[f(\hat{\mathbf{x}}_{n_0}^{\epsilon_2}, \xi)] \leq \hat{J}_{n_0}^{\epsilon_1}(\hat{\mathbf{x}}_{n_0}^{\epsilon_2}) + \epsilon_1) = 1$, and meanwhile guarantees the quality of the certificate $\hat{J}_{n_0}^{\epsilon_1}(\hat{\mathbf{x}}_{n_0}^{\epsilon_2})$ to be close to the optimal objective value of the Problem (P).

To achieve this, consider that the data set $\hat{\Xi}_n$ has been received. We then start by cheaply constructing a sequence of data-driven solutions $\hat{\mathbf{x}}^{(r)}$ with $r \geq r_n$, based on the data set $\hat{\Xi}_n$. After a finite number of iterations, if no new data has been received, the algorithm reaches $r = r_{n+1}$ such that $\hat{\mathbf{x}}^{(r_{n+1})} = \hat{\mathbf{x}}_n^{\epsilon_2}$ is ϵ_2-optimal, i.e., $J_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r_{n+1})})$ is $(\epsilon_1+\epsilon_2)$-close to $J_n^\star := \hat{J}_n(\hat{\mathbf{x}}_n^\star)$ with $\hat{\mathbf{x}}_n^\star \in \operatorname{argmin}_x \hat{J}_n(x)$. After a new data point is received, the algorithm finds the next ϵ_2-optimal data-driven solution $\hat{\mathbf{x}}_{n+1}^{\epsilon_2}$ and its certificate $\hat{J}_{n+1}^{\epsilon_1}(\hat{\mathbf{x}}_{n+1}^{\epsilon_2})$ with higher reliability $1-\beta_{n+1}$. In this way, online data can be assimilated over time while refining the constructed ϵ_2-optimal data-driven solutions $\{\hat{\mathbf{x}}_n^{\epsilon_2}\}_{n=1}^{\infty}$ with corresponding certificates $\{\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}_n^{\epsilon_2})\}_{n=1}^{\infty}$ that guarantee performance with high confidence $\{1-\beta_n\}_{n=1}^{\infty}$.

When the algorithm receives new data set $\hat{\Xi}_{n+1}$ before reaching to $r = r_{n+1}$, it safely starts from the current data-driven solution $\hat{\mathbf{x}}^{(r)}$. The algorithm then proceeds similarly on the data set $\hat{\Xi}_{n+1}$ by updating the subsequence index r_{n+1} to the current r.

Next, we focus on how to design the certificates based on the following assumption for f:

Assumption III.1 (Convexity-concavity and coercivity) *The known proper function $f : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$, $(\mathbf{x}, \xi) \mapsto f(\mathbf{x}, \xi)$ is continuously differentiable, convex in $\mathbf{x}$, concave in ξ and $f(\mathbf{x}, \tilde{\xi}) \to +\infty$ as $\|\mathbf{x}\| \to +\infty$ for all $\tilde{\xi} \in \mathbb{R}^m$.*

Certificate design via DRO theory: To design a certificate $\hat{J}_n(\hat{\mathbf{x}})$ for a given data-driven solution $\hat{\mathbf{x}}$, one can first use the data set $\hat{\Xi}_n$ from $\mathbb{P}$ to estimate an empirical distribution, $\hat{\mathbb{P}}^n$, and let $\mathbb{E}_{\hat{\mathbb{P}}^n}[f(\hat{\mathbf{x}}, \xi)]$ be the candidate certificate for the performance guarantee (1). However, such certificate only results in an approximation of the out-of-sample performance if $\mathbb{P}$ is unknown and (1) cannot be guaranteed in probability. Following [3], [5], we are to determine an *ambiguity set* $\hat{\mathcal{P}}_n$ containing all the possible probability distributions supported on $\mathcal{Z} \subseteq \mathbb{R}^m$ that can generate $\hat{\Xi}_n$ with high confidence. Then with the given feasible solution $\hat{\mathbf{x}}$, it is plausible to consider the worst-case expectation of the out-of-sample performance for all distributions contained in $\hat{\mathcal{P}}_n$. Such worst-case distribution offers an upper bound for the out-of-sample performance with high probability.

Denote by $\mathcal{M}_{\text{lt}}(\mathcal{Z}) \subset \mathcal{M}(\mathcal{Z})$ the set of light-tailed probability measures in $\mathcal{M}(\mathcal{Z})$, we make following assumption:

Assumption III.2 (Light tailed unknown distributions) *It is assumed that $\mathbb{P} \in \mathcal{M}_{\text{lt}}(\mathcal{Z})$, i.e., there exists an exponent $a > 1$ such that: $b := \mathbb{E}_{\mathbb{P}}[\exp(\|\xi\|^a)] < \infty$.*

Assumption III.2 validates the modern measure concentration result [13, Theorem 2] on $\mathcal{M}_{\text{lt}}(\mathcal{Z})$, which provides an intuition for considering the Wasserstein ball $\mathbb{B}_\epsilon(\hat{\mathbb{P}}^n)$ of center $\hat{\mathbb{P}}^n$ and radius ϵ as the ambiguity set $\hat{\mathcal{P}}_n$. Then equipped with the Wassenstein ball, we are able to provide the certificate that ensures the performance guarantee in (1) for any sequence of data-driven solutions $\{\hat{\mathbf{x}}^{(r)}\}_{r=1}^{\infty}$, by $\hat{J}_n(\hat{\mathbf{x}}^{(r)}) := \sup_{\mathbb{Q} \in \hat{\mathcal{P}}_n} \mathbb{E}_{\mathbb{Q}}[f(\hat{\mathbf{x}}^{(r)}, \xi)]$.

Worst-case distribution reformulation: To get the certificate $\hat{J}_n(\hat{\mathbf{x}}^{(r)})$, one needs to solve an infinite dimensional optimization problem, which is generally hard. Luckily, with an extended version of the strong duality results for moment problem [14, Lemma 3.4], we can reformulate the optimization problem for $\hat{J}_n(\hat{\mathbf{x}}^{(r)})$ into a finite-dimensional convex programming problem:

$$\begin{aligned} \hat{J}_n(\hat{\mathbf{x}}^{(r)}) := \max_{\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^m} & \frac{1}{n} \sum_{k=1}^n f(\hat{\mathbf{x}}^{(r)}, \hat{\xi}_k - \mathbf{y}_k), \\ \text{s.t.} \quad & \frac{1}{n} \sum_{k=1}^n \|\mathbf{y}_k\| \leq \epsilon(\beta_n), \end{aligned} \tag{P1$_n^{(r)}$}$$

where $\epsilon(\beta_n)$ is the radius of of $\mathbb{B}_{\epsilon(\beta_n)}$ as calculated in [12]. Given an ϵ_1-optimal solution $(\mathbf{y}_1^{\epsilon_1}, \ldots, \mathbf{y}_n^{\epsilon_1})$ of (P1$_n^{(r)}$), we denote a finite atomic probability measure at $\hat{\mathbf{x}}^{(r)}$ in $\mathbb{B}_{\epsilon(\beta_n)}$ by $\mathbb{Q}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)}) := \frac{1}{n}\sum_{k=1}^n \delta_{\{\hat{\xi}_k - \mathbf{y}_k^{\epsilon_1}\}}$. Then, $\mathbb{Q}_n^{\epsilon_1}$ is a worst-case distribution that can generate the data set $\hat{\Xi}_n$ with high probability (no less than $(1-\beta_n)$).

The concavity requirement in Assumption III.1 ensures that (P1$_n^{(r)}$) is a convex problem. Failure of Assumption III.1 may require us to find a relaxed problem of (P1$_n^{(r)}$) in order for efficiently generating $\hat{J}_n(\hat{\mathbf{x}}^{(r)})$ in the next section.

IV. CERTIFICATE GENERATION

Given the tolerance ϵ_1 and any feasible solution $\hat{\mathbf{x}}^{(r)}$, we present in this section the Certificate Generation Algorithm (CGA) for efficiently obtaining $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ and the ϵ_1 worst-case distribution, $\mathbb{Q}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ of an ϵ_1-proper data-driven solution $\hat{\mathbf{x}}^{(r)}$ over time, under the sequence of the data sets $\{\hat{\Xi}_n\}_{n=1}^N$. To achieve this, we first reformulate Problem (P1$_n^{(r)}$) to a convex optimization problem over a simplex. Then, we design the CGA to solve the customized problem to an ϵ_1-optimal solution efficiently.

For online implementation we have the following assumption on the computation of the gradient of the function f:

Assumption IV.1 (Cheap access of the gradients) *For any $\mathbf{x} \in \mathbb{R}^d$, the gradient of the function $h^{\mathbf{x}} : \mathbb{R}^m \to \mathbb{R}$ for $h^{\mathbf{x}}(\mathbf{y}) := f(\mathbf{x}, \mathbf{y})$ can be accessed cheaply.*

In the n^{th} time period with the data set $\hat{\Xi}_n$, we consider the following convex optimization problem over Δ_{2mn}:

$$\begin{aligned} \max_{\mathbf{v}\in\mathbb{R}^{2mn}} \ & \frac{1}{n}\sum_{k=1}^{n} h_k^r((A_n\mathbf{v})^{(k-1)m+1:km}), \\ \text{s.t.} \ & \mathbf{v} \in \Delta_{2mn}, \end{aligned} \tag{P2$_n^{(r)}$}$$

where for each $k \in \{1, \ldots, n\}$, $\hat{\xi}_k \in \hat{\Xi}_n$ and $\hat{\mathbf{x}}^{(r)} \in \mathbb{R}^d$, we define $h_k^r : \mathbb{R}^m \to \mathbb{R}$ as

$$h_k^r(\mathbf{y}) := f(\hat{\mathbf{x}}^{(r)}, \hat{\xi}_k - \mathbf{y}),$$

and the matrix $A_n := [\oplus_{i=1}^{n} I_m, -\oplus_{i=1}^{n} I_m] \in \mathbb{R}^{mn\times 2mn}$ where the first mn columns of A_n constitute the natural basis for the space $\mathbb{R}^{mn}$. The simplex is defined by $\Delta_{2mn} := \{\mathbf{v} \in \mathbb{R}^{mn} \mid \mathbf{1}_{2mn}{}^{\top}\mathbf{v} = n\epsilon(\beta_n),\ \mathbf{v} \geq 0\}$ and we denote by Λ_{2mn} the set of all the extreme points for the simplex Δ_{2mn}.

One can prove that solving (P1$_n^{(r)}$) is equivalent to solving (P2$_n^{(r)}$) for any feasible solution $\hat{\mathbf{x}}^{(r)}$ in every time period n, we refer to the online version [12] for details.

The Frank-Wolfe Algorithm variants, such as the Simplicial Algorithm [8] and the AFW algorithm [9], are known to be well suited for problems of the form (P2$_n^{(r)}$). The advantage of these is that they can handle the constraints of Problem (P2$_n^{(r)}$) via linear programming subproblems (LP) that result from the way in which the FW search point is found in [12, Section 2]. Intuitively, the following is done. For a number of iterations l, the following problems are solved alternatively:

$$\begin{aligned} \max_{\mathbf{v}\in\mathbb{R}^{2mn}} \ & \frac{1}{n}\sum_{k=1}^{n} \Big\langle \nabla h_k^r(\mathbf{y}_k^{(l-1)}), \cdots \\ & (A_n\mathbf{v})^{(k-1)m+1:km} - \mathbf{y}_k^{(l-1)} \Big\rangle, \\ \text{s.t.} \ & \mathbf{v} \in \Delta_{2mn}, \end{aligned} \tag{LP$^{(l)}$}$$

$$\begin{aligned} \max_{\gamma\in\mathbb{R}^{T+1}} \ & \frac{1}{n}\sum_{k=1}^{n} h_k^r(\sum_{i=0}^{T} \gamma^i \tilde{\mathbf{y}}_k^{(i)}), \\ \text{s.t.} \ & \gamma \in \Delta_T. \end{aligned} \tag{CP$^{(l)}$}$$

Notice that the search points generated for the linear subproblem (LP$^{(l)}$) at iteration l are the extreme points of the feasible set Δ_{2mn}. We denote by $I_n^{(l)}$ the set of these points. Considering the convex hull of $I_n^{(l)}$, parametrized by the convex combination coefficients γ of the points in $I_n^{(l)}$, an implicit feasible set $\text{conv}(I_n^{(l)})$ in a lower dimensional space can be constructed. Motivated by this, our Certificate Generation Algorithm iteratively solves the linear subproblem (LP$^{(l)}$), enlarges the implicit feasible set $\text{conv}(I_n^{(l)})$, and then searches a maximizer of the objective function of (P2$_n^{(r)}$) over $\text{conv}(I_n^{(l)})$ (represented as Δ_T in subproblem (CP$^{(l)}$)). This process is repeated to the next iteration $l+1$, and follows until an ϵ_1-optimal solution is found. Later we will see that the set $I_n^{(l)}$ allows to generate the certificate when assimilating data. We call this set the *candidate vertex set.*

For the above problems, notice that the subproblem (LP$^{(l)}$) maximizes a linear function over a simplex, therefore it is computationally cheap and an optimizer $\mathbf{v}^{(l)}$ is equivalently computed by choosing a sparse vector with only one positive entry, i.e., an extreme point of the feasible set of (LP$^{(l)}$), such that the nonzero component of $\mathbf{v}^{(l)}$ has the largest weight in the linear cost function of Problem (LP$^{(l)}$).

We refer to the online version of this paper [12] for complete description of the Certificate Generation Algorithm (denoted by [12, Algorithm 4]) and its finite convergence to achieve $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ and $\mathbb{Q}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ for any data-driven solution $\hat{\mathbf{x}}^{(r)}$ in the n^{th} time period with the data set $\hat{\Xi}_n$.

The worst-case computational bound of the Certificate Generation Algorithm at the iteration $l+1$, associated with the candidate solution $(\mathbf{y}_1^{(l)}, \ldots, \mathbf{y}_n^{(l)})$, is $\hat{J}_n(\hat{\mathbf{x}}^{(r)}) - \hat{J}_n^{\eta^{(l+1)}}(\hat{\mathbf{x}}^{(r)}) \leq 2mn\kappa^l\rho$, where $\kappa := 1 - \frac{\mu_f}{4C_f} \in (0,1) \subset \mathbb{R}$ is related to local strong convexity of f over Δ_{2mn}, and $\rho := \hat{J}_n(\hat{\mathbf{x}}^{(r)}) - \hat{J}_n^{\eta^{(1)}}(\hat{\mathbf{x}}^{(r)}) \leq \eta^{(1)}$ quantifies the initial distance of the objective function $\hat{J}_n$ and $\hat{J}_n^{\eta^{(1)}}$ at $\hat{\mathbf{x}}^{(r)}$. In other words, given the tolerance ϵ_1, in the worst case we need at least $l \geq \phi(n) := \log_\kappa(\frac{\epsilon_1}{2mn\rho})$ computational steps to find $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$.

However, how to generate $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ and $\mathbb{Q}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ online is unclear for each data-driven solution $\hat{\mathbf{x}}^{(r)}$. This is because that as the time period n moves, we need to not only obtain $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ and $\mathbb{Q}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ sufficiently fast, but also finding them by solving the Problem (P2$_n^{(r)}$) under a different data set $\hat{\Xi}_n$. As the size of $\hat{\Xi}_n$ grows, the dimension of the Problem (P2$_n^{(r)}$) increases. To deal these challenges, our Certificate Generation Algorithm exploits the relationships among the Problems (P2$_n^{(r)}$) with different data set $\hat{\Xi}_n$ by adapting the candidate vertex set $I_n^{(l)}$.

When the average data streaming rate is slower than the computational bound $\phi(1)$, we claim that Certificate Generation Algorithm can always find the certificate for each data set $\hat{\Xi}_n$. This is because in each time period n on average, we only have $2mn$ extreme points, and $2m(n-1)$ has been explored due to the adaptation of the candidate vertex set $I_n^{(0)}$. This indicates that in the worst-case situation the average data streaming rate should be lower than this value, in order to efficiently update the certificate for the sequence of the data-driven solutions.

V. An ϵ_2-optimal performance guarantee

In this section, we approach the construction of a sequence of the ϵ_2-optimal data-driven solutions $\{\hat{\mathbf{x}}_n^{\epsilon_2}\}_{n=1}^{\infty}$, associated with ϵ_2-lowest certificates $\{\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}_n^{\epsilon_2})\}_{n=1}^{\infty}$ over time, under the sequence of the data sets $\{\hat{\Xi}_n\}_{n=1}^{\infty}$. Specifically in the n^{th} time period, we start from $\hat{\mathbf{x}}^{(r)} := \hat{\mathbf{x}}^{(r_n)}$ with its associated ϵ_1-optimal certificate $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$, and as the iteration r grows, we are to find a sequence of $\hat{\epsilon}_1$-proper data-driven solutions, $\{\hat{\mathbf{x}}^{(r)}\}_{r=r_n}^{r_{n+1}}$, which converge to $\hat{\mathbf{x}}_n^{\epsilon_2}$ quickly and result in $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}_n^{\epsilon_2})$. We use a Subgradient Algorithm to obtain $\hat{\mathbf{x}}_n^{\epsilon_2}$, via a valid ϵ_1-subgradient of the certificate function $\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$, which denoted by $g_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ and can be computed as shown in the extended version [12].

However, for every time we generate a new data-driven solution $\hat{\mathbf{x}}^{(r+1)}$, the ϵ_1-optimal extreme distribution $\hat{\mathbb{Q}}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ associated with the last solution $\hat{\mathbf{x}}^{(r)}$ may not be a valid ϵ_1-optimal extreme distribution for $\hat{\mathbf{x}}^{(r+1)}$. To reduce the number of computations needed to obtain the new certificate for $\hat{\mathbf{x}}^{(r+1)}$, we denote by $g_n^{\epsilon^{(r)}}(\hat{\mathbf{x}}^{(r)})$ the $\epsilon^{(r)}$-subgradient at $\hat{\mathbf{x}}^{(r)}$, where $\epsilon^{(r)}$ may be greater than ϵ_1 for each r. Then by properly designing a sequence $\{\epsilon^{(r)}\}$, upper bounded by $\hat{\epsilon}_1$, and estimating the $\epsilon^{(r)}$-optimal extreme distributions, we will achieve a suboptimal proper data-driven solution efficiently.

Here, we employ the $\hat{\epsilon}_1$-Subgradient Algorithm with $\hat{\epsilon}_1 \gg \epsilon_1$, the divergent but square-summable step size rule, and scaled direction as follows: $\hat{\mathbf{x}}^{(r+1)} = \hat{\mathbf{x}}^{(r)} - \alpha^{(r)} \hat{g}_n^{\hat{\epsilon}_1}(\hat{\mathbf{x}}^{(r)}) / \max\{\|\hat{g}_n^{\hat{\epsilon}_1}(\hat{\mathbf{x}}^{(r)})\|\,,\, 1\}$. The estimated $\hat{\epsilon}_1$-subgradient $\hat{g}_n^{\hat{\epsilon}_1}(\hat{\mathbf{x}}^{(r)})$ at each iteration r is constructed and updated via the following considerations. Every time we generate the ϵ_1-optimal certificate from the Certificate Generation Algorithm at iteration r, the estimated $\hat{\epsilon}_1$-subgradient is constructed by $\hat{\mathbb{Q}}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ using lemma for the easy access of the subgradients [12], i.e., $g_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)}) \in \partial_{\hat{\epsilon}_1} \hat{J}_n(\hat{\mathbf{x}}^{(r)})$. During the execution of the $\hat{\epsilon}_1$-Subgradient Algorithm, we check for the $\hat{\epsilon}_1$-optimality of the certificate generated from $\hat{\mathbb{Q}}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$ at each subsequent iteration $\hat{r}$, using [12, Algorithm 3]. If the obtained suboptimality gap is such that $\eta > \hat{\epsilon}_1$ at $\hat{r} > r$, we generate a new ϵ_1-optimal distribution $\hat{\mathbb{Q}}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(\hat{r})})$ via Certificate Generation Algorithm and estimate the $\hat{\epsilon}_1$-subgradient using $\hat{\mathbb{Q}}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(\hat{r})})$. Otherwise, the certificate at $\hat{\mathbf{x}}^{(\hat{r})}$ is constructed using $\hat{\mathbb{Q}}_n^{\epsilon_1}(\hat{\mathbf{x}}^{(r)})$.

From the above construction, we see that each $\epsilon^{(r)}$, associated with a $\hat{g}_n^{\hat{\epsilon}_1}(\hat{\mathbf{x}}^{(r)})$, is such that $\epsilon^{(r)} \leq \hat{\epsilon}_1$. Then, we have the following lemma for the convergence of the $\hat{\epsilon}_1$-Subgradient Algorithm in the n^{th} time period.

Lemma V.1 (Convergence of the $\hat{\epsilon}_1$-Subgradient Algorithm to the ϵ_2-optimal solution given $\hat{\Xi}_n$) *In each time period n with an initial data-driven solution $\hat{\mathbf{x}}^{(r_n)}$, assume the subgradients defined by $\hat{g}_n^{\epsilon}(\hat{\mathbf{x}}^{(r)})$ are uniformly bounded, i.e., there exists a constant $L > 0$ such that $\|\hat{g}_n^{\epsilon}(\hat{\mathbf{x}}^{(r)})\| \leq L$ for all $r \geq r_n$ and $\epsilon \leq \hat{\epsilon}_1$. Let $\mu := \max\{L, 1\}$. Given a predefined $\epsilon_2 > 0$, and let the certificate tolerance ϵ_1 and the subgradient bound $\hat{\epsilon}_1$ such that $0 < \epsilon_1 \ll \hat{\epsilon}_1 < \epsilon_2/\mu$, then there exists a large enough number $\bar{r}$ such that the designed $\hat{\epsilon}_1$-Subgradient Algorithm has the following performance bounds:*

$$\min_{k \in \{r_n, \ldots, r\}} \{\hat{J}_n(\hat{\mathbf{x}}^{(k)})\} - \hat{J}_n(\hat{\mathbf{x}}_n^{\star}) \leq \epsilon_2, \quad \forall\, r \geq \bar{r},$$

and terminates at the iteration $r_{n+1} := \bar{r}$ with an ϵ_2-optimal solution under the data set $\hat{\Xi}_n$.

VI. Data Assimilation

We now describe a full algorithm to assimilate data online. The whole Online Data Assimilation Algorithm starts from some random initial data-driven solution. Then, for each given set of data points, we first generate its certificate via Certificate Generation Algorithm, after which an ϵ-proper data-driven solution is obtained, then we execute the Subgradient Algorithm to achieve a lower certificate. During the last set of iterations, the certificate may be lost and Certificate Generation Algorithm may have to be rerun again, and resume the Subgradient Algorithm after obtaining a valid certificate. If no data points come in, the algorithm terminates as soon as the Subgradient Algorithm terminates.

When there is streaming data, the algorithm needs to incorporate new data points every time they become available. Because of this, the feasible set of the Problem ($\text{P1}_n^{(r)}$) changes. This affects the dimension of Problem ($\text{P1}_n^{(r)}$), which grows by m, and results into an increase of the dimension of ($\text{LP}^{(l)}$) by $2m$. Second, the reliability increase from β_n to β_{n+1} results into a smaller radius $\epsilon(\beta_{n+1})$ of the Wasserstein ball $\mathbb{B}_{\epsilon(\beta_{n+1})}$.

Depending on the stage the new data point comes in, different strategies for generating initial point that is feasible for the new optimization problem can be considered. When data comes in during the execution of the ϵ-Subgradient Algorithm at iteration r, we use a current best $\hat{\epsilon}_1$-proper data-driven solution as the initial data-driven solution for the ϵ_2-optimal data-driven solution $\hat{\mathbf{x}}_{n+1}^{\epsilon_2}$, i.e., $\hat{\mathbf{x}}^{(r_{n+1})} := \hat{\mathbf{x}}_n^{\text{best}} \in \operatorname{argmin}_{k \in \{r_n, \ldots, r\}} \{\hat{J}_n(\hat{\mathbf{x}}^{(j)})\}$. The other initial data, such as $(\mathbf{y}_1^{(0)}, \ldots, \mathbf{y}_n^{(0)})$, $I_n^{(0)}$ and $\{\tilde{\mathbf{y}}_k^{(i)}\}_{i \in I_n^{(0)}}$, can be constructed following the same idea as when data point comes in during the execution of Certificate Generation Algorithm, the details of which are in [12]. By such scheme the online data can be assimilated into sequence of optimization problems [12, the Algorithm 5].

The Online Data Assimilation Algorithm has the anytime property, meaning that the performance guarantee is provided anytime, as soon as the first ϵ_1-proper data-driven solution is found. The algorithm then tries to make decisions that achieve lower certificates with higher reliability until we achieve the lowest possible certificate and guarantee the performance almost surely.

The transient behavior of the Online Data Assimilation Algorithm is naturally affected by the data streaming rate and the rate of convergence of intermediate algorithms (the assimilation rate). To further describe the effect of the data streaming rate, we call the data set stream $\{\hat{\Xi}_n\}_{n=1}^N$ *sufficiently slow in the n^{th} time period*, if we can find an $\hat{\mathbf{x}}_n^{\epsilon_2}$ in the $\hat{\epsilon}_1$-Subgradient Algorithm during the time period n. When the data streaming rate and assimilation rate are the same, and they are sufficiently slow for all n, the Online Data Assimilation Algorithm guarantees to find a low certificate with a good data-driven solution as established by the following finite convergence result.

Theorem VI.1 (Finite convergence of the Online Data Assimilation Algorithm) *Given any tolerance ϵ_1, ϵ_2, $\epsilon_3 > 0$ and sufficiently slow data streaming sets $\{\hat{\Xi}_n\}_{n=1}^{\infty}$. Then, there exists a large enough number $n_0(\epsilon_3) > 0$, such that the algorithm terminates in finite time with a sequence of ϵ_2-optimal ϵ_1-proper data-driven solutions $\{\hat{\mathbf{x}}_n^{\epsilon_2}\}_{n=1}^{n_0}$ associated with the sequence of the certificates $\{\hat{J}_n^{\epsilon_1}(\hat{\mathbf{x}}_n^{\epsilon_2})\}_{n=1}^{n_0}$ so that the performance guarantee (2) holds for all $n \leq n_0$. Moreover, we have a guaranteed ϵ_2-optimal and ϵ_1-proper data-driven solution $\hat{\mathbf{x}}_{n_0}^{\epsilon_2}$ and a certificate $\hat{J}_{n_0}^{\epsilon_1}(\hat{\mathbf{x}}_{n_0}^{\epsilon_2})$ such that the performance guarantee holds almost surely, i.e.,*

$$\mathbb{P}^{\infty}(\mathbb{E}_{\mathbb{P}}[f(\hat{\mathbf{x}}_{n_0}^{\epsilon_2}, \xi)] \leq \hat{J}_{n_0}^{\epsilon_1}(\hat{\mathbf{x}}_{n_0}^{\epsilon_2}) + \epsilon_1) = 1,$$

and meanwhile the quality of the designed certificate $\hat{J}^{\epsilon_1}_{n_0}(\hat{\mathbf{x}}^{\epsilon_2}_{n_0})$ *is guaranteed, i.e., for all the rest of the data sets* $\{\hat{\Xi}_n\}_{n=n_0}^{\infty}$*, any element in the desired certificate sequence* $\{\hat{J}^{\epsilon_1}_{n}(\hat{\mathbf{x}}^{\epsilon_2}_{n})\}_{n=n_0}^{\infty}$ *satisfies*

$$\sup_{n\geq n_0} \hat{J}^{\epsilon_1}_{n}(\hat{\mathbf{x}}^{\epsilon_2}_{n}) \leq J^\star + \epsilon_1 + \epsilon_2 + \epsilon_3,$$

where $J^\star$ *is the optimal objective value for* (P).

VII. Simulation Results

In this section, we demonstrate the application of the Online Data Assimilation Algorithm to find an ϵ-proper data-driven solution $x \in \mathbb{R}^{30}$ for Problem (P). We consider $N = 50$ iid sample points $\{\hat{\xi}_k\}_{k=1}^{N}$ streaming randomly in between every 1 to 3 seconds with each data point $\hat{\xi}_k \in \mathbb{R}^{10}$ a realization of the unknown distribution $\mathbb{P}$. Here, we assume that the unknown distribution is a mixture of the multivariate uniform distribution on $[-2, 2]^{10}$ and the multivariate normal distribution $\mathcal{N}(2.5 \cdot \mathbf{1}_{10}, 4 \cdot I_{10})$. We assume the cost function $f : \mathbb{R}^{30} \times \mathbb{R}^{10} \to \mathbb{R}$ to be $f(\mathbf{x}, \xi) := \mathbf{x}^\top A\mathbf{x} + \mathbf{x}^\top B\xi + \xi^\top C\xi$ with random values for the positive semi-definite matrix $A \in \mathbb{R}^{30\times30}$, $B \in \mathbb{R}^{30\times10}$ and negative definite matrix $C \in \mathbb{R}^{10\times10}$. Let the reliability $1 - \beta_n := 1 - 0.95e^{1-\sqrt{n}}$ and use the parameter $c_1 = 2$, $c_1 = 1$ to design the radius $\epsilon(\beta_n)$ of the Wasserstein ball. We sample the initial data-driven solution $\hat{\mathbf{x}}^{(0)}$ from the uniform distribution $[0, 10]^{30}$. The tolerance for the algorithm is $\epsilon_1 = 10^{-5}$, $\epsilon_2 = 10^{-6}$, and $\epsilon_3 = 10^{-6}$.

To evaluate the quality of the obtained ϵ-proper data-driven solution with the streaming data, we estimate the optimizer of (P), $\mathbf{x}^\star$, by minimizing the average value of the cost function f for a validation data set with $N_{\text{val}} = 10^4$ data points randomly generated from the distribution $\mathbb{P}$ (in the simulation case $\mathbb{P}$ is known). We take the resulting objective value as the estimated optimal objective value for Problem (P), i.e., $J^\star := \hat{J}^\star(\mathbf{x}^\star)$. We calculate $\hat{J}^\star(\mathbf{x}^\star)$ using the underline distribution $\mathbb{P}$, serving as the true but unknown scale to evaluate the goodness of the certificate obtained throughout the algorithm.

Figure 1 shows the evolution of the certificate sequence $\{\hat{J}^{\epsilon_1}_{n}(\hat{\mathbf{x}}^{(r)})\}_{n=1,r=1}^{N,\infty}$ for the decision sequence $\{\hat{\mathbf{x}}^{(r)}\}_{r=1}^{\infty}$. The blue line in the Figure 1 shows the relative goodness of the certificates for the currently used ϵ_1-proper data-driven solution $\hat{\mathbf{x}}^{(r)}$ calibrated by the estimated optimal value $J^\star$ over time. The red points indicate that a new certificate $\hat{J}^{\epsilon_1}_{n+1}(\hat{\mathbf{x}}_{(r)}(t))$ is processing when the new data set is incorporated, while at these time intervals the old certificate $\hat{J}^{\epsilon_1}_{n}(\hat{\mathbf{x}}^{\epsilon_2}_{n})$, associated with the ϵ_2-optimal and ϵ_1-proper data-driven solution $\hat{\mathbf{x}}^{\epsilon_2}_{n}$, is still valid to guarantee the performance under the old reliability β_n. This situation commonly happens when a new data set $\hat{\Xi}_{n+1}$ is streamed in and a new certificate $\hat{J}^{\epsilon_1}_{n+1}(\hat{\mathbf{x}}_{(r)}(t))$ is yet to be obtained. It can be seen that after a few samples streamed, the obtained certificate becomes close to the estimated true optimal value $J^\star$ within the 10% range.

VIII. Conclusions

In this paper, we have proposed the Online Data Assimilation Algorithm for real-time data-driven solutions of (P) with guaranteed out-of-sample performance. Such solutions are available any time during the execution of the algorithm, and the optimal data-driven solution are approached with a (sub)linear convergence rate. The algorithm terminates after collecting sufficient amount of data to make a good decision. Future work will generalize the results for weaker assumptions of the problem and potentially extend the algorithm to scenarios that include system dynamics.

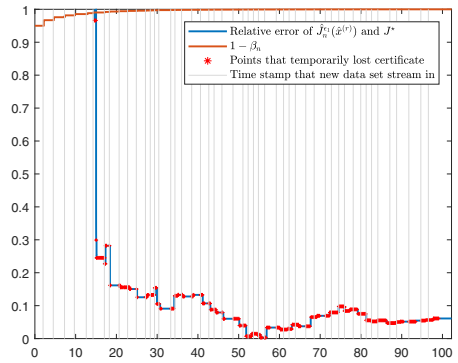

Fig. 1: Relative goodness of the certificates for the performance guarantee of an ϵ_1-proper data-driven solution. The x-axis is time (seconds) and the y-axis plots the relative goodness function $R(t) := |(\hat{J}_n(\hat{\mathbf{x}}^{(r)}(t)) - J^\star)/J^\star|$.

References

[1] A. Shapiro, D. Dentcheva *et al.*, *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2014, vol. 16.

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

[3] A. Cherukuri and J. Cortés, "Data-driven distributed optimization using wasserstein ambiguity sets," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2017, pp. 38–44.

[4] ——, "Cooperative data-driven distributionally robust optimization," *arXiv preprint https://arxiv.org/pdf/1711.04839.pdf*, 2017.

[5] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: performance guarantees and tractable reformulations," *Mathematical Programming*, pp. 1–52, 2017.

[6] R. Gao and A. J. Kleywegt, "Distributionally robust stochastic optimization with wasserstein distance," *arXiv preprint arXiv:1604.02199*, 2016.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[8] C. Holloway, "An extension of the Frank and Wolfe method of feasible directions," *Mathematical Programming*, vol. 6, no. 1, pp. 14–27, 1974.

[9] S. Lacoste-Julien and M. Jaggi, "On the global linear convergence of Frank-Wolfe optimization variants," in *Advances in Neural Information Processing Systems*, 2015, pp. 496–504.

[10] D. P. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex analysis and optimization*. Athena Scientific, 2003.

[11] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[12] D. Li and S. Martínez, "Online data assimilation in distributionally robust optimization," UC San Diego, Tech. Rep., 2018, online extended version at http://fausto.ucsd.edu/sonia/papers/2018-DL-SM.pdf and in arxiv under the same title (math OC, 2201766).

[13] N. Fournier and A. Guillin, "On the rate of convergence in wasserstein distance of the empirical measure," *Probability Theory and Related Fields*, vol. 162, no. 3-4, pp. 707–738, 2015.

[14] A. Shapiro, *On Duality Theory of Conic Linear Problems*. Boston, MA: Springer US, 2001, pp. 135–165. [Online]. Available: https://doi.org/10.1007/978-1-4757-3403-4_7