Online optimization and data assimilation with performance guarantees

D. Li¹ and S. Martínez¹

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 \( \xi \) under an unknown distribution \( \mathcal{P} \). To deal with this, we aim to devise a procedure that incorporates samples of \( \xi \) sequentially and adjusts decisions using the real-time data. We approach this problem in a distributionally robust optimization framework and propose a novel Online Data Assimilation Algorithm for this purpose. This algorithm guarantees out-of-sample performance of decisions with 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 a sufficiently slow rate of streaming data, and provide a criteria for the termination of the algorithm after a certain number of data have been collected. Simulations illustrate the results.

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, 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 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 guarantees with high probability. Then, the out-of-sample performance of the data-driven decision is obtained as the worst-case ambiguity set, which contains the true distribution of the data-generating system with high probability. The out-of-sample performance of the data-driven decision is obtained under a sufficiently slow rate of streaming data, and provide a criteria for the termination of the algorithm after a certain number of data have been collected. Simulations illustrate the results.

Literature Review: Optimization under uncertainty is a popular research area, and as such available methods include stochastic optimization [2] and robust optimization [3].

Recently, data-driven distributionally robust optimization has regained popularity thanks to its out-of-sample performance guarantees, see e.g. [4], [5] and [6], [7], for a distributed algorithm counterpart, 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 decision 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 [8], \( \phi \)-divergence [9] and the Wasserstein distance [4]. Here, following the paper [6], 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 [4] and in [6] do not consider the update of the data-driven decision over time, which serves as the focus of this work. In terms of the algorithm design, our work connects to various convex optimization methods [10] such as the Frank-Wolfe (FW) Algorithm (e.g., conditional gradient algorithm), the Subgradient Algorithm, and their variants, see e.g. [11]–[13] and references therein. Our emphasis on the convergence of the data-driven decision obtained through a sequence of optimization problems contrasts with typical algorithms developed for single (non-updated) problems.

Statement of Contributions: In this paper, we propose a new Online Data Assimilation Algorithm to solve decision-making problems subject to uncertainty. The distribution of the uncertainty is unknown and thus the algorithm adjusts decisions based on realizations of \( \xi \) that are revealed and collected sequentially over time. The new algorithm addresses four challenges: 1) the evaluation of the out-of-sample performance of every possible online decision; 2) the adaptation to online, increasingly-larger data sets to reach a decision with out-of-sample performance guarantees with increasingly higher probabilities; 3) the availability of an online decision vector with performance guarantees at any time; 4) the capability of handling sufficiently large streaming data sets.

To address 1), we start from a distributionally robust optimization (DRO) problem setting. This leads to a worst-case optimization over an ambiguity set or neighborhood of the empirical distribution constructed from a data set. To solve this intractable problem, we reformulate it into an equivalent
convex optimization over a simplex. This enables us to explore
the simplex vertex set and find a performance certificate for the
decision with a given confidence. When the data is streaming
online, we consider a sequence of DRO problems and their
equivalent convex problems employing increasingly larger data
sets. Thus, as the data streams, the associated problems are
deﬁned over simplices of increasingly larger dimension. The
similarities of these feasible sets allow us to assimilate the
online data via specialized Frank-Wolfe Algorithm variants,
thus solving 2) via a CERTIFICATE GENERATION AL-
GORITHM described in Section IV. Further, to seek for decisions
that approach to the minimizers of the optimization problem,
the ONLINE DATA ASSIMILATION Algorithm adapts its
iterations online via a Subgradient Algorithm as described in
Section V. We show in Section VI that the resulting ONLINE
DATA ASSIMILATION Algorithm is ﬁnitely convergent in
the sense that the conﬁdence of the out-of-sample performance
guarantee for the generated data-driven decision converges to 1
as the number of data samples increases to a suﬃciently large
but ﬁnite value. Under this scheme, a data-driven decision with
certain performance guarantee is also available any time as
soon as the algorithm ﬁnishes generating the ﬁrst certiﬁcate
for the initial decision, which resolves the challenge 3). To ex-
pedite the algorithm and deal with the challenge 4), we devel-
oped in Section VII an incremental covering algorithm to process
the streaming data set and obtain low-dimensional ambiguity
sets in DRO problems. These new sets are based on a weighted
iterations online via a Subgradient Algorithm as described in
Section VII. We deﬁne the domain of the function \( f \rightarrow (\tilde{\lambda}) \rightarrow \Omega \) projects the set \( \tilde{\lambda} \) onto the set \( \Upsilon \) under the Euclidean norm.

Notations from Probability Theory: Let \( (\Omega, \mathcal{F}, \mathbb{P}) \) be a
probability space, with \( \Omega \) the sample space, \( \mathcal{F} \) a σ-algebra on \( \Omega \),
and \( \mathbb{P} \) the associated probability distribution. Let \( \xi : \Omega \rightarrow \mathbb{R}^m \)
be an induced multivariate random variable. We denote by
\( \mathcal{Z} \subseteq \mathbb{R}^m \) the support of the random variable \( \xi \) and denote by \( \mathcal{M}(\mathcal{Z}) \) the space of all probability distributions supported
on \( \mathcal{Z} \) with ﬁnite ﬁrst moment. In particular, \( \mathbb{P} \in \mathcal{M}(\mathcal{Z}) \).
To measure the distance between distributions in \( \mathcal{M}(\mathcal{Z}) \), in
this paper we use the dual characterization of the Wasserstein
metric \( d_{W} : \mathcal{M}(\mathcal{Z}) \times \mathcal{M}(\mathcal{Z}) \rightarrow [0, \infty) \), deﬁned by
\[
d_{W}(Q_1, Q_2) := \sup_{f \in \mathcal{L}} \int_{\mathcal{Z}} f(\xi)Q_1(d\xi) - \int_{\mathcal{Z}} f(\xi)Q_2(d\xi),
\]
where \( \mathcal{L} \) is the space of all Lipschitz functions deﬁned on \( \mathcal{Z} \)
with Lipschitz constant 1. A closed Wasserstein ball of radius \( \omega \) centered at a distribution \( \mathbb{P} \in \mathcal{M}(\mathcal{Z}) \) is denoted by
\( \mathbb{B}_{\omega}(\mathcal{P}) := \{ Q \in \mathcal{M}(\mathcal{Z}) \mid d_{W}(\mathbb{P}, Q) \leq \omega \} \).

Numerical Optimization Methods: There are mainly two
types of Numerical Optimization methods that serve as the
main ingredients of our ONLINE DATA ASSIMILATION AL-
GORITHM. One type is given by Frank-Wolfe Algorithm (FWA)
variants and another is the Subgradient Algorithm. In this
subsection, we describe FWA and the Away-step Frank-Wolfe
Algorithm (AFWA) for the sake of completeness. We will
combine it with another variant, the Simplicial Algorithm,
in Section IV. For the Subgradient Algorithm, please refer
to [15]-[20].

The Frank-Wolfe Algorithm over a unit simplex. To solve
convex programs over a unit simplex, we introduce the FWA
and AFWA following [12], [13]. Let us denote the m-

In the m-dimensional Euclidean space, the m-dimensional non-

Notations: Let \( \mathbb{R}^m, \mathbb{R}^{m \times n} \) and \( \mathbb{R}^{m \times d} \) denote respectively the

This section introduces notations, including some from
Probability Theory that help describe the distributionally
robust optimization framework following [4], and a brief sum-
mary of the numerical methods employed next.

The Frank-Wolfe Algorithm over a unit simplex. To solve
convex programs over a unit simplex, we introduce the FWA
and AFWA following [12], [13]. Let us denote the m-
dimensional Euclidean ball centered at \( x \in \mathbb{R}^m \) with radius \( \omega \) as
the set (\( B_{\omega}(x) := \{ y \in \mathbb{R}^m \mid \| y - x \| \leq \omega \}. \)
Matrices \( A_1 \in \mathbb{R}^{m \times d} \) and \( A_2 \in \mathbb{R}^{p \times q} \), we let \( A_1 \oplus A_2 \) 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 \( \text{conv}(I) \)
indicate its convex hull. The gradient of a real-valued function
\( f : \mathbb{R}^m \rightarrow \mathbb{R} \) is written as \( \nabla_f(x) \). The \( j \)th component of the
gradient vector is denoted by \( \nabla_f(x) \). We use \( \text{dom} f \) to
defer the domain of the function \( f \), i.e., \( \text{dom} f := \{ x \in
\mathbb{R}^m \mid -\infty < f(x) < +\infty \}. \) We call the function \( f \) proper
if \( \text{dom} f \neq \emptyset \). We say a function \( F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \) is
convex-concave on \( \mathcal{X} \times \mathcal{Y} \) if, for any point \( (\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}, \)
\( x \mapsto F(x, \bar{y}) \) is convex and \( y \mapsto F(\bar{x}, y) \) is concave. We refer
to this property as \( F \) being convex-concave in \( (x, y) \). We use the
notation \( \text{sgn} : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto \{ -1, 0, 1 \} \) denote the sign
function. Finally, the projection operator \( \text{proj}_X(\lambda) : \mathcal{X} \rightarrow \mathcal{Y} \) projects
the set \( \lambda \) onto the set \( \Upsilon \) under the Euclidean norm.
The classical FWA then iteratively finds a FW direction and solves a line search problem over this direction until an \( \epsilon \)-optimal solution \( x^\epsilon := x^{(k)} \) is found, certified by \( \eta^{(k)} := -\nabla f(x^{(k)})^T d_r^{(k)} \leq \epsilon \).

It is known that the classical FWA has linear convergence rate if the cost function \( f \) is \( \mu \)-strongly convex and the optimum is achieved in the relative interior of the feasible set \( \Delta_n \). If the optimal solution lies on the boundary of \( \Delta_n \), then this algorithm only has sublinear convergence rate, due to the zig-zagging phenomenon [13]. AFWA is an extension of the FWA that guarantees the linear convergence rate of the problem \( (*) \) under some conditions related to the local strong convexity. The main difference between AFWA and the classical FWA is that the latter solves the line-search problem after obtaining a descent direction by considering all points, while the AFWA chooses a descend direction that prevents zig-zagging. We summarize the convergence properties of the AFWA here. For complete descriptions of the AFWA, we refer the reader to [13] and an older version of this paper [21].

### Theorem II.1 (Linear convergence of AFWA [13, Theorem 8])

Suppose the function \( f \) has a constant curvature \( C_f \) and a geometric strong convexity constant \( \mu_f \) on \( \Delta_n \), as defined in [13, page 17-18]. Let us define the decay rate \( \kappa := 1 - \mu_f / (4C_f) \in (0, 1) \subset \mathbb{R} \). Then the suboptimality bound at the iteration point \( x^{(k)} \) of the AFWA decreases geometrically as \( f(x^{(k+1)}) - f(x^\star) \leq \kappa (f(x^{(k)}) - f(x^\star)) \).

### III. Problem Description

Consider a decision-making problem of the form

\[
\min_{x \in \mathbb{R}^d} \mathbb{E}_P[f(x, \xi)],
\]

where \( x \in \mathbb{R}^d \) is the decision variable, the random variable \( \xi : \Omega \to \mathbb{R}^m \) is induced by the probability space \( (\Omega, F, P) \), and the expectation of \( f \) is taken w.r.t. the unknown distribution \( P \in \mathcal{M}(Z) \). We aim to develop an ONLINE DATA ASSIMILATION ALGORITHM that efficiently adapts iterations for decisions \( x \) of (P) with online streaming data. In this section, we describe the algorithm procedure employing the DRO terminology in [4, 6].

Let \( \{x^{(r)}\}_{r=1}^{\infty} \) be a sequence where, for each iteration \( r \), the decision \( x^{(r)} \) is feasible for (P). The ONLINE DATA ASSIMILATION ALGORITHM generates \( \{x^{(r)}\}_{r=1}^{\infty} \) while sequentially collecting iid realizations of the random variable \( \xi \) under \( P \), denoted by \( \xi_n, \ n = 1, 2, \ldots \). This defines a sequence of streaming data sets, \( \Xi_n \subseteq \Xi_{n+1} \), for each \( n \). W.l.o.g. assume that each \( \Xi_{n+1} \) consists of just one more new data point, i.e., \( \Xi_{n+1} = \Xi_n \cup \{\xi_{n+1}\} \) and \( \Xi_1 = \{\xi_1\} \). The time between updates of \( \Xi_n \) and \( \Xi_{n+1} \) corresponds to a certain time period, referred to as the \( n \)-th time period. The subsequence of decisions obtained during this period is labeled by \( \{x^{(r)}\}_{r=\tau_n+1}^{\tau_{n+1}} \).

The objective of our algorithm is to make real-time decisions for (P) that have a potentially low objective value, while assimilating information from the \( \Xi_n \).

To quantify the quality of the decisions \( \{x^{(r)}\}_{r=1}^{\infty} \), we introduce the following terms. We call an \( x^{(r)} \in \mathbb{R}^d \) a proper data-driven decision of (P), if \( x^{(r)} \) is feasible and its out-of-sample performance, defined by \( \mathbb{E}_P[f(x^{(r)}, \xi)] \), satisfies the following performance guarantee:

\[
P^n(\mathbb{E}_P[f(x^{(r)}, \xi)] \leq J_n(x^{(r)})) \geq 1 - \beta_n,
\]

where the certificate \( J_n(x^{(r)}) \) is a function that indicates the goodness of \( x^{(r)} \) under the data set \( \Xi_n \). If \( x^{(r)} \) is adopted during the \( n \)-th time period, then \( \mathbb{E}_P[f(x^{(r)}, \xi)] \leq J_n(x^{(r)}) \) is an event that depends on the \( n \) samples in \( \Xi_n \), and \( P^n \) denotes the probability with respect to these. The confidence \( 1 - \beta_n \in (0, 1) \subset \mathbb{R} \) governs the choice of \( x^{(r)} \) and the resulting certificate \( J_n(x^{(r)}) \). Finding an approximate certificate is much easier than finding an exact one. Based on this, we call \( x^{(r)} \) \( \epsilon_x \)-proper, if it satisfies (1) with \( J_n(x^{(r)}) \) such that \( J_n(x^{(r)}) \leq J_n(x^{(r)}) + \epsilon_1 \). The certificates \( J_n(x^{(r)}) \) and their approximates \( J_n(x^{(r)}) \) provide an upper bound to the optimal value of (P) with high confidence.

In each time period \( n \), given a confidence level \( 1 - \beta_n \), our goal is to approach to an \( \epsilon_x \)-proper data-driven decision with a low certificate. Thus, we call any proper data-driven decision \( \epsilon_x \)-optimal, labeled as \( x^{(r)} \), if \( J_n(x^{(r)}) \leq J_n(x^{(r)}) \leq \epsilon_2 \) for all \( x \in \mathbb{R}^d \). Then, for any \( \epsilon_2 \)-optimal and \( \epsilon_x \)-proper data-driven decision \( x^{(r)} \) with certificate \( J_n(x^{(r)}) \) and \( \epsilon_1 \ll \epsilon_2 \), we have the following performance guarantee:

\[
P^n(\mathbb{E}_P[f(x^{(r)}, \xi)] \leq J_n(x^{(r)})) + \epsilon_1 \geq 1 - \beta_n.
\]
set \( \Xi_{n+2} \) at iteration \( r < n+2 \), it safely starts generating \( \{x^{(r)}\}_{r=n+2}^{n+3} \) by letting \( x^{(n+2)} \) to be the current best decision and finding its \( J_{n+2}^{\ast}(x^{(n+2)}) \). Then the algorithm proceeds similarly on the data set \( \Xi_{n+2} \).

In this way, online data can be assimilated while refining the \( \{x^{(r)}\}_{r=n}^{N} \) with certificates \( \{J_{n}^{\ast}(x^{(r)})\}_{n=1}^{N} \) that guarantee performance with high confidence \( \{1 - \beta_{n}\}_{n=1}^{N} \).

Next, we focus on how to design \( J_{n} \) based on the assumption that \( f \) is continuous. To do this, we employ ideas from DRO. The material in the next two subsections is adapted from [4], [6], [7].

Certificate design: To find certificates, one can first use the data \( \Xi_{n} \) to estimate an empirical distribution, \( \hat{P}_{n} \), and let \( E_{\hat{P}_{n}}[f(x, \xi)] \) be the candidate certificate for the performance guarantee (1). More precisely, assume that the data set \( \Xi_{n} \) is uniformly sampled from \( P \). The discrete empirical probability measure associated with \( \Xi_{n} \) is the following:

\[
\hat{P}_{n} := \frac{1}{n} \sum_{k=1}^{n} \delta_{\hat{\xi}_{k}},
\]

where \( \delta_{\hat{\xi}_{k}} \) is a Dirac measure at \( \hat{\xi}_{k} \). The candidate certificate is

\[
J_{n}^{\text{emp}}(x) := E_{\hat{P}_{n}}[f(x, \xi)] = \frac{1}{n} \sum_{k=1}^{n} f(x, \hat{\xi}_{k}).
\]

The above approximation \( \hat{P}_{n} \) of \( P \), also known as the sample-average estimate, makes \( J_{n}^{\text{emp}} \) easy to compute. However, such value only results in an approximation of the out-of-sample performance if \( P \) is unknown. Following [4], [6], we are to determine an ambiguity set \( \mathcal{P}_{n} \) containing all the possible probability distributions supported on \( Z \subseteq \mathbb{R}^{m} \) that can generate \( \Xi_{n} \) with high confidence. Then, with the given feasible decision \( x \), it is plausible to consider the worst-case expectation of the out-of-sample performance for all distributions contained in \( \mathcal{P}_{n} \). The solution to such problem offers an upper bound for the out-of-sample performance with high probability in form of (1), and we refer to this upper bound as the certificate of decision \( x \).

In order to quantify the certificate for an \( \epsilon_{1} \)-proper data-driven decision, we denote by \( \mathcal{M}_{B}(Z) \subset \mathcal{M}(Z) \) the set of light-tailed probability measures in \( \mathcal{M}(Z) \), and introduce the following assumption for \( P \):

**Assumption III.1 (Light tailed unknown distributions)** It holds that \( P \in \mathcal{M}_{B}(Z) \), i.e., there exists an exponent \( a > 1 \) such that:

\[
b := E_{P}[\exp([\|\xi\|]^{a})] < \infty.
\]

Assumption III.1 validates the following modern measure of concentration result, which provides an intuition for considering the Wasserstein ball \( \mathcal{B}(\hat{P}_{n}) \) of center \( \hat{P}_{n} \) and radius \( \epsilon \) as the ambiguity set \( \mathcal{P}_{n} \).

**Theorem III.1 (Measure concentration [22, Theorem 2])** If \( P \in \mathcal{M}_{B}(Z) \), then

\[
P^{n}\left\{ d_{W}(\hat{P}_{n}, P^{n}) \geq \epsilon \right\} \leq \left\{ \begin{array}{ll}
c_{1}e^{-c_{2}n\epsilon^{2}m}, & \text{if } \epsilon \leq 1, \\
(1 + n)^{\epsilon / 2}c_{2}e^{-c_{2}n\epsilon^{2}}, & \text{if } \epsilon > 1,
\end{array} \right.
\]

for all \( n \geq 1 \), \( m \neq 2 \), and \( \epsilon > 0 \), where \( c_{1}, c_{2} \) are positive constants that only depend on \( m, a, \) and \( b \).

Then, equipped with this result, we are able to provide the certificate that ensures the performance guarantee in (1), for any decision \( x \in \mathbb{R}^{d} \).

**Lemma III.1 (Certificate in Performance Guarantee (1))** Given \( \Xi_{n} := \{\xi_{k}\}_{k=1}^{n}, \beta_{n} \in (0, 1) \) and \( x^{(r)} \in \mathbb{R}^{d} \) for any \( r \geq n_{r} \). Let

\[
\epsilon(\beta_{n}) := \left\{ \begin{array}{ll}
\left( \log(c_{1}b^{-1}) \right)^{1 / \max(2, m)}, & \text{if } n \geq \frac{\log(c_{1}b^{-1})}{c_{2}} \\
\left( \log(c_{1}b^{-1}) \right)^{1 / n}, & \text{if } n < \frac{\log(c_{1}b^{-1})}{c_{2}}
\end{array} \right.,
\]

and \( \mathcal{P}_{n} := \mathcal{B}(\epsilon(\beta_{n}))(\hat{P}_{n}) \). Then the following certificate satisfies the performance guarantee in (1) for all \( x^{(r)} \in \mathbb{R}^{d} \):

\[
J_{n}(x^{(r)}) := \sup_{Q \in \mathcal{P}_{n}} E_{Q}[f(x^{(r)}, \xi)].
\]

Worst-case distribution reformulation: To get the certificate in (5), one needs to solve an infinite dimensional optimization problem. Luckily, problem (5) can be reformulated into a finite-dimensional convex problem as follows.

**Theorem III.2 (Convex reduction of (5) [4, Application of Theorem 4.4])** Under Assumption III.1 on the light tailed distribution of \( P \), for all \( \beta_{n} \in (0, 1) \) the value of the certificate in (5) for the data-driven decision \( x^{(r)} \) under the data set \( \Xi_{n} \) is equal to the optimal value of the following optimization problem:

\[
J_{n}(x^{(r)}) := \sup_{y_{1}, \ldots, y_{n} \in \mathbb{R}^{n}} \frac{1}{n} \sum_{k=1}^{n} f(x^{(r)}, \hat{\xi}_{k} - y_{k}),
\]

subject to

\[
\frac{1}{n} \sum_{k=1}^{n} \|y_{k}\| \leq \epsilon(\beta_{n}).
\]

where \( \epsilon(\beta_{n}) \) is the radius of \( \mathcal{B}(\epsilon(\beta_{n})) \) calculated from (4). Moreover, given any feasible point \( y^{(l)} := (y_{1}^{(l)}, \ldots, y_{n}^{(l)}) \) of (P1)\(_{n}^{(r)} \), indexed by \( l \), define a finite atomic probability measure at \( x^{(r)} \) in the Wasserstein ball \( \mathcal{B}(\epsilon(\beta_{n})) \) of the form:

\[
Q_{l}^{(r)}(x^{(r)}) := \frac{1}{n} \sum_{k=1}^{n} \delta_{\hat{\xi}_{k} - y_{k}^{(l)}}.
\]

Now, denote by \( Q^{r}_{\ast}(x^{(r)}) \) the distribution in (6) constructed by an optimizer \( y^{*} := (y_{1}^{*}, \ldots, y_{n}^{*}) \) of (P1)\(_{n}^{(r)} \) and evaluated over data-driven decisions \( x^{(r)} \). Then, \( Q^{r}_{\ast} \) is a worst-case distribution that can generate the data set \( \Xi_{n} \) with high probability (no less than \( 1 - \beta_{n} \)).

IV. Certificate Generation

Given the tolerance \( \epsilon_{1} \), sequentially available data sets \( \{\Xi_{n}\}_{n=1}^{N} \) and decisions \( \{x^{(r)}\}_{r=1}^{\infty} \), we present in this section the CERTIFICATE GENERATION Algorithm for approximated certificates \( \{J_{n}^{\ast}(x^{(r)})\}_{n=1}^{\infty, r} \) and resulting \( \epsilon_{1} \)-worst-case distributions \( \{Q_{n}^{r}(x^{(r)})\}_{n=1}^{\infty, r} \). To achieve this, we first reformulate Problem (P1)\(_{n}^{(r)} \) to Problem (P2)\(_{n}^{(r)} \), a convex optimization
problem over a simplex. Then, we design the CERTIFICATE GENERATION ALGORITHM (Algorithm 1) to solve (P2_n) to an approximated certificate \( J_n^*(\mathbf{x}(r)) \) efficiently. Finally, we analyze the convergence of Algorithm 1 under \( (\Xi_n)_n^{\infty} \).

Let us consider the \( n \)th time period, the data set \( \Xi_n \), and the sequence \( \{\mathbf{x}(r)\} \). In this period, for each \( \mathbf{x}(r) \) and \( \xi_k \in \Xi_n \), with \( k \in \{1, \ldots, n\} \), we define a parametrized function

\[
h_k^*(\mathbf{y}) := f(\mathbf{x}(r), \xi_k - \mathbf{y}).
\]

For each \( \mathbf{x}(r) \), let us consider the following convex optimization problem over a simplex:

\[
J_n(\mathbf{x}(r)) := \max_{u_k, v_k \in \mathbb{R}^m} \frac{1}{n} \sum_{k=1}^{n} h_k^*(u_k - v_k),
\]

s.t. \( (u, v) \in n\epsilon(\beta_n)\Delta_{2mn}, \)

where the concatenated variable \((u, v)\) is composed of \( u := (u_1, \ldots, u_n) \) and \( v := (v_1, \ldots, v_n) \); and the scalar \( n\epsilon(\beta_n) \) regulates the size of the feasible set via scaling of the unit simplex \( \Delta_{2mn} := \{(u, v) \in \mathbb{R}^{2mn} \mid 1_{2mn}^T (u, v) = 1, u \geq 0, v \geq 0\} \). We denote by \( \Delta_{2mn}^\ast \) the set of all the extreme points for the simplex \( n\epsilon(\beta_n)\Delta_{2mn} \).

The following lemma shows that Problem \( (P1_n) \) and Problem \( (P2_n) \) are equivalent for \( J_n^*(\mathbf{x}(r)) \) and \( Q_n^*(\mathbf{x}(r)) \).

**Lemma IV.1 (Equivalence of the problem formulation)** Let \( \mathbf{x}(r) \) be a feasible decision on the \( n \)th time period with the data set \( \Xi_n \). Then solving \( (P1_n) \) is equivalent to solving \( (P2_n) \) in the sense that

1. For any feasible solution \((\tilde{u}, \tilde{v})\) of \( (P2_n) \), let \( \tilde{y} := \tilde{u} - \tilde{v} \). Then \( \tilde{y} \) is feasible for \( (P1_n) \).
2. For any feasible solution \( \tilde{y} \) of \( (P1_n) \), there exists a feasible point \((\tilde{u}, \tilde{v})\) of \( (P2_n) \).
3. Assume that the point \((\tilde{u}, \tilde{v})\) is an optimizer of \( (P2_n) \). Then by letting \( \tilde{y}^* := \tilde{u}^* - \tilde{v}^* \), the point \( \tilde{y}^* \) is also an optimizer of \( (P1_n) \), with the same optimal value.

To obtain \( \{J_n^*(\mathbf{x}(r))\}_{n,r} \) and \( \{Q_n^*(\mathbf{x}(r))\}_{n,r} \) by solving \( (P2_n^0) \), we develop the CERTIFICATE GENERATION ALGORITHM in I via Frank-Wolfe Algorithm variants, e.g., the Simplicial Algorithm [12] and the AFWA as in Section II.

The algorithm proceeds iteratively at times \( l = 1, 2, \ldots \). At each iteration \( l \), \((\mathbf{u}(l), \mathbf{v}(l))\) is the candidate optimizer of \( (P2_n^0) \). Let the objective value of \( (P2_n^0) \) at \((\mathbf{u}(l), \mathbf{v}(l))\) be \( J_n^l(\mathbf{x}(r)) \) and, equivalently, write the candidate optimizer in form of \( \mathbf{y}(l) := \mathbf{u}(l) - \mathbf{v}(l) \) (exploiting the equivalence in Lemma IV.1). Each candidate \( \mathbf{y}(l) \) is associated with it a set of FW search points denoted by \( I_n^l := \{y(i) := \mathbf{u}(i) - \mathbf{v}(i), i \in \{1, \ldots, T\}\} \). As we will see later, the set \( I_n^l \) plays the role of generating the certificate when approximating data, and is called the candidate vertex set.

Algorithm 1 alternatively solves the following problems:

\[
\max_{\mathbf{u}_k, \mathbf{v}_k \in \mathbb{R}^m} \frac{1}{n} \sum_{k=1}^{n} \left< \nabla h_k^*(y_k^{(l-1)}), \cdots, \mathbf{u}_k - \mathbf{v}_k - y_k^{(l-1)} \right>, \quad (Lb_n^l)
\]

s.t. \( (\mathbf{u}, \mathbf{v}) \in n\epsilon(\beta_n)\Delta_{2mn}, \)

\[
\max_{\gamma \in \mathbb{R}^T} \frac{1}{n} \sum_{k=1}^{n} \sum_{i=0}^{T} \gamma_i h_k^*(y_k^{(i)}), \quad (Cf_n^l)
\]

s.t. \( \gamma \in \Delta_T \).

Note that the candidate optimizer \( y_k^{(l-1)} \) parameterizes the linear program (LP_n^l). The solution to (LP_n^l) is then used to refine the set of FW search points \( I_n^l = \{y(i)\} \), which span the implicit feasible set \( \Delta_T \equiv \text{conv}(I_n^l) \) in problem (CF_n^l). A solution to (CF_n^l) then determines the new candidate optimizer \( y_l \) of the LP problem at the next iteration.

**Algorithm 1**

1. \( l \leftarrow 0; \)
2. Update \( y_l, I_n^l, T \) and \( \gamma^l; \)
3. repeat
4. \( l \leftarrow l + 1; \)
5. \((\Omega_l, \eta_l) \leftarrow \text{LP}(\mathbf{x}(r), \Xi_n, y_l^{(l-1)}); \)
6. \( I_n^l \leftarrow I_n^{(l-1)} \cup \Omega_l; \)
7. \((\gamma^l, J_n^l(\mathbf{x}(r))) \leftarrow \text{AFWA}(Cf_l^l); \)
8. \( y_l \leftarrow \sum_{i=0}^{T} \gamma_i y_l^{(i)}, y_l^{(i)} \in I_n^l \) for each \( i \);
9. until \( \eta_l \leq \epsilon_l \)
10. return \( J_n^l(\mathbf{x}(r)) = J_n^l(\mathbf{x}(r)), y_l^* = y_l, Q_n^*(\mathbf{x}(r)) = \frac{1}{n} \sum_{k=1}^{n} \delta(\xi_k - y_l^*). \)

More precisely, at each iteration \( l = 1, 2, \ldots \), with a candidate optimizer \( y_l^{(l-1)} \), Algorithm 1 first solves subproblem (LP_n^l) using the point search algorithm (Algorithm 2), which returns the optimal objective value \( \eta_l \) and the set of maximizers \( \Omega_l \) such that \( \eta_l \geq J_n^l(\mathbf{x}(r)) - J_n^{(l-1)}(\mathbf{x}(r)) \) and \( \Omega_l \subset \Delta_{2mn} \). In particular, Algorithm 2 computes all optimizers by iteratively choosing a sparse vector with only a positive entry. That is, an extreme point of the feasible set of \( (LP_n^l) \), such that the nonzero component of \((\tilde{u}^{(l)}, \tilde{v}^{(l)})\) has the largest absolute gradient component in the linear cost function of subproblem (LP_n^l). As a result, Algorithm 2 returns the value \( \eta_l \) that certifies the \( \epsilon_l \)-suboptimality condition to the optimal objective of Problem \( (P2_n^0) \) and the set \( \Omega_l \) that updates the candidate vertex set to \( I_n^l := I_n^{(l-1)} \cup \Omega_l \). Let us denote the cardinality of the set \( I_n^l \) by \( T := |I_n^l| \). Then for each \( i \in \{1, \ldots, T\} \), we use \( y_l^{(i)} := \tilde{u}^{(i)} - \tilde{v}^{(i)} \) to denote the \( i \)th candidate vertex in \( I_n^l \). Using the obtained \( I_n^l \) the algorithm solves the Problem (CF_n^l) over the simplex \( \Delta_T := \{\gamma \in \mathbb{R}^T \mid 1_T^T \gamma = 1, \gamma \geq 0\} \), where each component \( \gamma_i \) of \( \gamma \in \Delta_T \) represents the convex combination coefficient of the candidate vertex \( y_l^{(i)} \). After solving (CF_n^l) to \( \epsilon_l \)-optimality via the AFWA, an \( \epsilon_l \)-optimal weighting \( \gamma^l \in \Delta_T \) with the objective value \( J_n^l(\mathbf{x}(r)) \) is obtained. A new candidate
optimizer \( y^{(l)} \) is then calculated by \( y^{(l)} = \sum_{i=0}^{L} \epsilon_{i} y^{(l)} \). The algorithm repeats the process and increments \( l \) if the optimality gap \( \eta^{(l)} \) is greater than \( \epsilon_{1} \), otherwise it returns the certificate \( J_{n}^{\epsilon_{1}}(x^{(l)}) := J_{n}^{(l)}(x^{(l)}) \), an \( \epsilon_{1} \)-optimal solution \( y^{(l)} := y^{(l)} \) and an \( \epsilon_{1} \)-worst-case distribution \( Q_{n}^{\epsilon_{1}}(x^{(l)}) := \frac{1}{n} \sum_{k=1}^{n} 1{\{\xi_{k} - y^{(l)}\}} \).

**Algorithm 2** Point search LP \((x^{(l)}, \Xi_{n}, y^{(l-1)})\)

1. Set \( \Omega^{(l)} := \emptyset \).
2. Let \( H := \{j(k) | j \in \{1, \ldots, m\}, k \in \{1, \ldots, n\}\} \).
3. Let \( S^{(l)} := \text{argmax}_{j(k) \in H} \{\pm \nabla h_{j}^{*}(y^{(l-1)})\} \).
4. While \( S^{(l)} \neq \emptyset \), do:
   1. Pick \((h, l) \in S^{(l)}\) and let \( \tilde{y} = 0_{mn} \).
   2. Update \( y_{h,l} \leftarrow n\epsilon_{0} \text{sign}(\nabla h_{h}^{*}(y^{(l-1)})) \).
   3. Update \( \Omega^{(l)} \leftarrow \Omega^{(l)} \cup \{\tilde{y}\} \).
   4. Update \( S^{(l)} \leftarrow S^{(l)} \setminus \{(h, l)\} \).
   5. Pick any \( \tilde{y} \in \Omega^{(l)} \) and,
   6. Set \( \eta^{(l)} = \frac{1}{n} \sum_{k=1}^{n} \{\nabla h_{k}^{*}(y_{k}^{(l-1)}), \tilde{y}_{k} - y_{k}^{(l-1)}\} \).
7. Return the set \( \Omega^{(l)} \) and the optimality gap \( \eta^{(l)} \).

Adapting Algorithm 1 to online data sets \( \Xi_{n} \) is inherently difficult due to the changes in the Problems \( P^{(r)}_{2n} \). The size of \( \Xi_{n} \) grows by 1, the dimension of the Problem \( P^{(r)}_{2n} \) increases by 2m. To obtain \( J_{n}^{\epsilon_{1}}(x^{(r)}) \) and \( Q_{n}^{\epsilon_{1}}(x^{(r)}) \) sufficiently fast, we exploit the relationship among Problems \( P^{(r)}_{2n} \), for different \( n \), by adapting the candidate vertex set \( I_{n}^{(r)} \). Specifically, we initialize the set \( I_{n}^{(r)} \) for the new Problem \( P^{(r)}_{2n+1} \) by \( I_{n+1}^{(r)} \), constructed from the previous \( P^{(r)}_{2n} \). Suppose that the Certificate Generation Algorithm receives a new data set \( \Xi_{n+1} \supseteq \Xi_{n} \) at some intermediate iteration \( l \) with candidate vertex set \( I_{n}^{(l)} \). At this stage, the subset Conv \( I_{n}^{(l)} \) has been explored by the previous optimization problem, and the gradient information of the objective function based on the data set \( \Xi_{n} \) has been partially integrated. Then, by projecting the set \( I_{n}^{(l)} \) onto the set of extreme points of the new Problem \( P^{(r)}_{2n} \), i.e., \( I_{n}^{(r)} := \text{proj}_{\Delta_{m^{2n+1}}}^{\Xi_{2n+1}}(\{\{y^{(r)}, 0_{m}\} | y^{(r)} \in I_{n}^{(l)}\}) \), the subset Conv \( I_{n+1}^{(r)} \) is already explored. Such integration contributes to the reduction of the number of iterations in the Algorithm 1 for Problems \( P^{(r)}_{2n} \). This insight gives us a sense of the worst-case efficiency to update a certificate under the streaming data.

For the online implementation of Algorithm 1, we have the following assumption on the local strong concavity of the function \( f \) and the computation of the gradient of \( f \):

**Assumption IV.1 (Local strong concavity)** For any feasible decision \( \hat{x} \in \{x^{(r)}\}_{r=1}^{\infty} \) and any data point \( \xi \in \Xi_{N} \), the function \( h : \mathbb{R}^{m} \rightarrow \mathbb{R}, y \mapsto f(\hat{x}, \xi - y) \) is differentiable, has a constant curvature \( C_{h} \), and has a positive geometric strong concavity constant \( \mu_{h} \) on \( \Delta_{2m^{N}} \).

**Assumption IV.2 (Cheap access of the gradients)** For any feasible decision \( \hat{x} \in \{x^{(r)}\}_{r=1}^{\infty}, \) we denote by \( \nabla h(y) \) the gradient of the function \( h : \mathbb{R}^{m} \rightarrow \mathbb{R}, y \mapsto f(\hat{x}, y) \) and assume it can be accessed cheaply.

The concavity requirement of \( h \) ensures that \( P^{(r)}_{2n} \) is a convex problem. Under Assumption IV.2, we show the convergence properties of Algorithm 1.

**Theorem IV.1 (Convergence of the Certificate Generation Algorithm 1)** Let the tolerance \( \epsilon_{1} \) and any feasible decision \( x^{(r)} \) be given. Let us choose \( y^{(0)} = 0_{m} \) and \( I_{0}^{(r)} = \emptyset \) to be the initial candidate optimizer and candidate vertex set for Algorithm 1, respectively. Consider the online data sets \( \Xi_{n} \) and the set of parameterized functions \( \{h_{n}\}_{n=1}^{\infty} \). Assume Assumption IV.1 and Assumption IV.2 hold. Then, for each data set \( \Xi_{n} \), there exists a parameter \( \kappa \in (0, 1) \subset \mathbb{R} \) such that the worst-case computational bound \( \phi(n) \) of Algorithm 1, depending on \( \kappa \), is

\[
\phi(n) \leq (2mn)\log_{\kappa} \left( \frac{\epsilon_{1}}{J_{n}(x^{(r)}) - J_{n}^{(r)}(x^{(r)})} \right)
\]

Moreover, consider that data sets \( \Xi_{n} \) are streaming online. Then there exists a parameter \( \tilde{\kappa} \in (0, 1) \subset \mathbb{R} \) and a computational bound

\[
\tilde{\phi}(n) := (2mn)\log_{\tilde{\kappa}} \left( \frac{\epsilon_{1}}{J_{n}(x^{(r)}) - J_{n}^{(r)}(x^{(r)})} \right)
\]

such that, if the average data-streaming rate is slower than \( \phi(1) \), then Algorithm 1 is guaranteed to obtain the certificates \( \{J_{n}^{\epsilon_{1}}(x^{(r)})\}_{n=1}^{\infty} \) and \( \{Q_{n}^{\epsilon_{1}}(x^{(r)})\}_{n=1}^{\infty} \).

V. AN \( \epsilon_{2} \)-OPTIMAL PERFORMANCE GUARANTEE

In this section, we aim to construct a subsequence of \( \epsilon_{2} \)-optimal data-driven decisions \( \{x^{(r)}\}_{r=1}^{\infty} \), associated with the \( \epsilon_{2} \)-lowest certificates \( \{J_{n}^{\epsilon_{2}}(x^{(r)})\}_{n=1}^{\infty} \) over time. We achieve this by means of: the Subgradient Algorithm to derive an \( \epsilon_{1} \)-proper decision sequence \( \{x^{(r)}\}_{r=1}^{\infty} \), and concatenation of \( \{x^{(r)}\}_{r=1}^{\infty} \) to obtain a \( \{x^{(r)}\}_{r=1}^{\infty} \).

To construct \( \epsilon_{1} \)-proper decision sequence \( \{x^{(r)}\}_{r=1}^{\infty} \) on the \( n \)th time period with set \( \Xi_{n} \), let us consider the following problem:

\[
J_{n}^{*} := \inf_{x \in \mathbb{R}^{d}} J_{n}(x),
\]

where the function \( J_{n}(x) \) is defined as in either (5) or (P1\((r)\)), and we assume the approximation of \( J_{n}(x) \) to be evaluated as in Section IV.

To solve this Problem to \( J_{n}^{*}(x^{(r)}_{n}) \), we have the following assumption on the convexity of \( f \):

**Assumption V.1 (Convexity in \( x \))** The function \( f_{\xi} : \mathbb{R}^{d} \rightarrow \mathbb{R}, x \mapsto f(x, \xi) \) is convex for all \( \xi \in \mathbb{R}^{m} \).

Assumption V.1 results in convexity of \( J_{n}(x) \) as follows.

**Lemma V.1 (Convexity of \( J_{n}(x) \))** If Assumption V.1 holds, then for each \( n \in \{1, \ldots, N\} \) the certificate \( J_{n}(x) \) defined by (5) is convex in \( x \).

Lemma V.1 allows us to apply the Subgradient Algorithm [15], [16], [19] to obtain \( x^{(r)}_{n} \) via \( \{x^{(r)}_{n}\}_{r=1}^{\infty} \) and the following lemma.
Lemma V.2 (Easy estimate of the ϵ-subgradients of Jn(x))
Let the tolerance ϵ1 and time period n be given. For any feasible decision we denote an ϵ1-optimal solution and ϵ1-worst-case distribution of (P1n(1)) by yf1 and Qf1n(x(r)), respectively. Let us consider the function g'_n = Qf1n(x(r)) defined as

\[ g'_n(x) := \frac{d}{dx} E_{Qf1n(x(r))}[f(x, \xi)]. \]

We define an ϵ-subdifferential of Jn(x) at x, by ∂xJn(x).

Then, for all \( \epsilon \geq \epsilon_1 \) we have the following:

\[ g'_n(x(r)) \in \partial xJ_n(x(r)), \]

or equivalently, for every \( z \in \text{dom} J_n \) and \( \epsilon \geq \epsilon_1 \), we have

\[ J_n(z) \geq J_n(x(r)) + g'_n(x(r))^T (z - x(r)) - \epsilon. \]

Moreover, for any \( \tilde{x} \in \mathbb{R}^d \), there exist \( \eta > 0 \) such that for all \( \epsilon \geq \eta \) the following relation holds:

\[ g'_n(\tilde{x}) \in \partial xJ_n(\tilde{x}). \]

Using the previous lemma, every time we achieve an ϵ1-proper decision x(r) from Algorithm 1, a valid ϵ1-subgradient \( g'_n(x(r)) \) of the certificate function can be computed. Thus, the Subgradient Algorithm can be employed to reach an ϵ1-proper data-driven decision with a lower certificate.

To do this, we make use of the divergent but square-summable step size rule and scaled direction, as follows:

\[ x^{(r+1)} = x^{(r)} - \alpha^{(r)} \frac{g'_n(x^{(r)})}{\max\{\|g'_n(x^{(r)})\|, 1\}}, \tag{7} \]

where \( \alpha^{(r)}>0 \), \( \sum_{r=1}^{\infty} \alpha^{(r)} = +\infty \) and \( \sum_{r=1}^{\infty} (\alpha^{(r)})^2 < +\infty \).

The Subgradient Algorithm requires access of \( \{g'_n\}_{r=r_1}^{r=n} \), which are obtained from Algorithm 1. To reduce the number of computations, we estimate the candidate subgradient functions \( \{g'_n\}_{r=r_1}^{r=n} \), as follows. Let \( \epsilon_{SA} \gg \epsilon_1 \) be a specified tolerance. At some iteration \( r \geq r_n \), assume that an \( \epsilon_1 \)-optimizer \( y_{-1} \) and \( \epsilon_1 \)-worst-case distribution \( Q_{f1n}(x^{(r)}) \) are obtained from Algorithm 1. Using \( Q_{f1n}(x^{(r)}) \), we calculate the function g'_n at x^{(r)} and perform the subgradient iteration (7). At iteration \( r+1 \) with x^{(r+1)}, we firstly check for the suboptimality of Problem (P1_{r+1}(x^{(r+1)})) using initial candidate optimizer \( y^{(0)} := y_{-1}^{(r)} \) in Algorithm 2. If the optimality gap \( y^{(r)} \) is less than \( \epsilon_{SA} \), we estimate the candidate subgradient function g'_n+1 using g'_n and proceed the subgradient iteration. Otherwise, we obtain g'_n+1 from Algorithm 1, which is again an \( \epsilon_1 \)-subgradient function at x^{(r+1)}. We cheaply construct a sequence of \( \epsilon_{SA} \)-subgradient functions \( \{g'_n\}_{r=r_1}^{r=n} \), that achieve an \( x^{(r)} \) efficiently.

The following lemma follows from the convergence of the Subgradient Algorithm applied to our problem scenario.

Lemma V.3 (Convergence of ϵSA-Subgradient Algorithm)
In each time period n with an initial data-driven decision x(r), assume the subgradients defined in Lemma V.2 are uniformly bounded, i.e., there exists a constant L > 0 such that \( \|g'_n\| \leq L \) for all r ≥ r_1.

Given a predefined \( \epsilon_2 > 0 \), let the certificate tolerance \( \epsilon_1 \) and the subgradient tolerance \( \epsilon_{SA} \) be such that \( 0 < \epsilon_1 < \epsilon_{SA} < \epsilon_2 / \mu \). Let \( \mu := \max\{L, 1\} \). Then there exists a large enough number \( r \) such that the above designed Subgradient Algorithm in (7) has the following performance bounds:

\[ \min_{k \in \{r_n, \ldots, r\}} \{J_n(x(k))\} - J_n(x^*) \leq \epsilon_2, \quad \forall r \geq r, \]

and terminates at the iteration \( r_{n+1} := r \) with an \( \epsilon_2 \)-optimal decision under the data set \( \Xi_n \) by \( x^{(n+1)} \in \text{argmin}_{k \in \{r_n, \ldots, r\}} \{J_n(x(k))\} \).

To quantify the effect of subgradient estimation on convergence rate under \( \Xi_n \), we have the following theorem.

Theorem V.1 (Worst-case computational bound for an \( \epsilon_{SA} \))
In each time period n with an initial \( x^{(r-1)} \), let us consider the algorithm setting as in Lemma V.3. Then, there exist parameters \( \kappa \in (0, 1) \), \( t > \epsilon_1 \) such that the computational steps \( \varphi(n, r) \) to reach \( x^{(r)} \) are bounded by

\[ \varphi(n, r) \leq \phi(n) + r \left( \log_2 \left( \frac{2}{t} \right) + 1 \right), \]

where \( r \) are the subgradient steps of Lemma V.2. The value \( \phi(n) \) is the worst-case computational bound as in Theorem IV.1 and one should use \( \phi(1) \) in the bound in place of \( \phi(n) \) if considering a data-streaming scenario.

VI. DATA ASSIMILATION
This section summarizes and analyzes our ONLINE DATA ASSIMILATION ALGORITHM for online data \( \{\Xi_n\}_{n=1}^{N} \). Specifically, we present the algorithm procedure, its transient behavior and the convergence result.

The ONLINE DATA ASSIMILATION ALGORITHM starts from some random initial decision \( x^{(r)} \in \mathbb{R}^d \) and a data set \( \Xi_n \), with \( r = 0 \) and \( n = 1 \). Then, we first generate its certificate \( J_{-1}^{(1)}(x^{(r)}) \) via Algorithm 1, then execute the Subgradient Algorithm to achieve decisions \( \{x^{(r+1)}, x^{(r+2)}, \ldots\} \) with lower and lower certificates \( \{J_{-1}^{(r)}(x^{(r+1)}), J_{-1}^{(r)}(x^{(r+2)}), \ldots\} \). This algorithm has the anytime property, meaning that the performance guarantee is provided anytime, as soon as the first \( \epsilon_1 \)-proper data-driven decision with certificate \( J_{-1}^{(r)}(x^{(r)}) \) is found. If no new data set \( \Xi_{n+1} \) comes in, the algorithm terminates as soon as the Subgradient Algorithm terminates at iteration \( r_{n+1} \). Otherwise, the algorithm then tries to make decisions using more data, which achieves lower certificates with higher confidence until we obtain the lowest possible certificate and guarantee the performance almost surely. The details of the whole algorithm procedure are summarized in the Algorithm 3.
Algorithm 3 ONLINE DATA ASSIMILATION ALGORITHM.

Require: Goes to Step 3 whenever $\Xi_n \leftarrow \Xi_{n+1}$.
1: Set $\epsilon_1, \epsilon_2, \epsilon_{\delta A}, \Xi_1, x^{(0)} \in \mathbb{R}^d$, $y^{(0)} = 0_m$ and $J_1^{(0)} = \emptyset$;
2: $n \leftarrow 1$, $r \leftarrow 0$;
3: $r_n \leftarrow r$;
4: $(J_n^{(r)}(x^{(r')}), y^{r}, Q_n^{(r)}(x^{(r'}))) \leftarrow$ Algorithm 1;
5: repeat
6: $x^{(r+1)} \leftarrow (x^{(r')}, g^{r})$ as in (7), $r \leftarrow r + 1$;
7: if $\eta > \epsilon_{\delta A}$, then
8: Go to Step 4;
9: else
10: Update $g^{r} \leftarrow g^{r-1}$;
11: if $J_n^{(r)}(x^{(r')}) < J_n^{(r)}(x^{\text{best}})$, then
12: Update and post $(x^{\text{best}}, J_n^{(r)}(x^{\text{best}}))$;
end if
13: end if
14: until $\|x^{(r)} - x^{(r-1)}\| < \epsilon_2$;
15: $r_{n+1} \leftarrow r$;
16: Post $x^{r^2} := x^{\text{best}}, J_n^{(r^2)}(x^{r^2}) := J_n^{(r)}(x^{\text{best}})$;
17: Wait for $\Xi_{n+1}$, or Termination if $n = n_0$.

The transient behavior of the ONLINE DATA ASSIMILATION ALGORITHM is affected by the data streaming rate and the rate of convergence of intermediate algorithms (assimilation rate). To further describe these effects, we call the data stream $\Xi_n$ 

\begin{align*}
\text{efficiently slow in the} \ n\text{th time period, if we can find an} \ x_n^{r^2} \ \text{in the Subgradient Algorithm during the time period} \ n, \ \text{where its worst-case assimilation rate is described in Theorem VI.1. Further, we call} \ \{\Xi_n\}_{n=1}^N \ \text{slow in the} \ n\text{th time period if we can find at least one certificate during time period} \ n, \ \text{where its worst-case assimilation rate is described in IV.1. When the data streaming rate is not faster than the worst-case assimilation rate as in Theorem VI.1, the ONLINE DATA ASSIMILATION ALGORITHM guarantees to find} \ \{x_n^{r^2}\}_{n=1}^N. \ \text{When the data streams sufficiently slow for at least one time period, it guarantees an} \ x_n^{r^2}. \ \text{When the data streaming rate is slow for at least one time period, the ONLINE DATA ASSIMILATION ALGORITHM guarantees to find a} \ J_n^{(r)} \ \text{for an} \ x^{(r)}. \ \text{When the data stream are not slow for all time periods, the ONLINE DATA ASSIMILATION ALGORITHM will hold on the newly streamed data set, to make the data streaming rate sufficiently slow and achieve a better data-driven decision efficiently.}
\end{align*}

Next, we state the convergence result of the ONLINE DATA ASSIMILATION ALGORITHM when the data streams are sufficiently slow for all the time periods.

Theorem VI.1 (Finite convergence of the ONLINE DATA ASSIMILATION ALGORITHM) Given any tolerance $\epsilon_1, \epsilon_2 > 0$ and sufficiently slow data streaming sets $\{\Xi_n\}_{n=1}^N$ with $N < \infty$, Then, the ONLINE DATA ASSIMILATION ALGORITHM guarantees to find a sequence of $\epsilon_2$-optimal $\epsilon_1$-proper data-driven decisions $\{x_n^{r^2}\}_{n=1}^N$ associated with the sequence of the certificates $\{J_n^{(r)}(x_n^{r^2})\}_{n=1}^N$ so that the performance guarantee (2) holds for all $n$.

In addition, given any tolerance $\epsilon_3$ and sufficiently slow data streams with $N \rightarrow \infty$, there exists a large enough number $n_0(\epsilon_3) > 0$, such that the algorithm terminates in finite time with a guaranteed $\epsilon_2$-optimal and $\epsilon_1$-proper data-driven decision $x_n^{r_0}$ and a certificate $J_n^{(r_0)}(x_n^{r_0})$ such that the performance guarantee holds almost surely, i.e.,

$$\mathbb{P}^{n_0}(E_\mathbb{P}[f(x_n^{r_0}, x)] \leq J_n^{(r_0)}(x_n^{r_0}) + \epsilon_1) = 1, \quad (8)$$

and meanwhile the quality of the designed certificate $J_n^{(r_0)}(x_n^{r_0})$ is guaranteed, i.e., for all the rest of the data sets $\{\Xi_n\}_{n=n_0}^\infty$, any element in the desired certificate sequence $\{J_n^{(r_0)}(x_n^{r_0})\}_{n=n_0}^\infty$ satisfies

$$\sup_{n \geq n_0} J_n^{(r_0)}(x_n^{r_0}) \leq J^* + \epsilon_1 + \epsilon_2 + \epsilon_3, \quad (9)$$

where $J^* := \inf \mathbb{E}_\mathbb{P}[f(x, \xi)]$ is the optimal objective value for the original unsolvable problem (P).

VII. DATA INCREMENTAL COVERING

In this section, we aim to handle large streaming data sets for efficient ONLINE DATA ASSIMILATION ALGORITHM. To achieve this, we firstly propose an incremental covering algorithm (ICA). This algorithm leverage the pattern of the data points for a new ambiguity set, denoted by $\mathcal{P}_n$. Then, we adapt $\mathcal{P}_n$ for a variant of ONLINE DATA ASSIMILATION ALGORITHM. The resulted algorithm enables us to construct subproblems which have the dimension lower than that generated without ICA, and we verify its capability of handling large data sets in simulation.

Incremental covering algorithm (ICA): Let $\zeta$ and $\omega$ denote the center and radius of the Euclidean ball $B_\omega(\zeta)$, respectively. For each data set $\Xi_n$ and a given $\omega$, let $\mathcal{C}_n \subset \Xi_n$, denote the set of points such that $\Xi_n \subset \bigcup_{\zeta \in \mathcal{C}_n} B_\omega(\zeta)$. Let $p := |\mathcal{C}_n|$ denote the number of these Euclidean balls. To account for the number of data points that are covered by a specific ball, we associate each ball $B_\omega(\zeta_k)$ a weighting parameter $\theta_k$. Denote by $Q_n := \{\theta_k\}_k^{p}$ the set of these parameters. Then, as data sets $\{\Xi_n\}_{n=1}^N$ are sequentially accessible, we are to incrementally cover data sets by adapting $\mathcal{C}_n$ and $Q_n$.

Formally, ICA works as follows. Let $\mathcal{C}_0 = \emptyset$ and $Q_0 = 0$. For the $n$th time period with set $\Xi_n$, we first initialize sets by setting $\mathcal{C}_n := \mathcal{C}_{n-1}$ and $Q_n := Q_{n-1}$. To generate a random cover for $\Xi_n$, we randomly and sequentially evaluate each newly streamed data point. Let $\zeta \in \Xi_n \setminus \Xi_{n-1}$ denote the data point under consideration. If $\zeta \notin B_\omega(\zeta_k)$ for all $\zeta_k \in \mathcal{C}_n$, we update $\mathcal{C}_n \leftarrow \mathcal{C}_n \cup \{\zeta_{p+1} := \zeta\}$, $Q_n \leftarrow Q_n \cup \{\theta_{p+1} := 1\}$ and $p = |\mathcal{C}_n|$. If $\zeta$ is covered by some (at least one) Euclidean balls, i.e., $\zeta \in B_\omega(\zeta_k)$ for some $k$ with $\zeta_k \in \mathcal{C}_n$, we only update $Q_n$. Let $\ell_k$ denote the number of the balls that cover $\zeta$ and let $I_k \subset \{1, \ldots, p\}$ denote the index set of these balls. Then we update elements of $Q_n$ via $\theta_k \leftarrow \theta_k + \ell_k$ for all $k \in I_k$. After all the new data points are evaluated in this way, we achieve a cover of $\Xi_n$. Then, as the data set streams over time, the algorithm incrementally updates the cover and weights. By construction, we see that $|\mathcal{C}_n| \leq n$.

Next, we use $\mathcal{C}_n$ and $Q_n$ to construct a new ambiguity set that results in potentially low dimensional subproblems in the ONLINE DATA ASSIMILATION ALGORITHM.
Design of the ambiguity set $\hat{\mathcal{P}}_n$: Following ICA, we consider a distribution $\hat{p}^n$ associated with $\Xi_n$, as follows:

$$\hat{p}^n := \frac{1}{n} \sum_{k=1}^{p} \theta_k \delta_{\xi_k},$$

where $\delta_{\xi_k}$ is a Dirac measure at the center of the covering ball $B_\omega(\xi_k)$ and $\theta_k$ is the associated weight of $B_\omega(\xi_k)$. We claim the distribution $\hat{p}^n$ is close to the empirical distribution $P^n$ under the Wasserstein metric, using the following lemma.

**Lemma VII.1 (Distribution $\hat{p}^n$ is a good estimate of $P^n$)**

Let the radius $\omega$ of the Euclidean ball be chosen. Then the distribution $\hat{p}^n$ constructed by the incremental covering algorithm on $\Xi_n$ is close to $P^n$ under the Wasserstein metric, i.e., $d_W(\hat{p}^n, P^n) \leq \omega$.

Then equipped with Lemma VII.1 and Theorem III.1 on the measure of concentration result, we can provide the certificate that ensures the performance guarantee in (1).

**Lemma VII.2 (Tractable certificate generation for $x$ with Performance Guarantee (1) using $\hat{p}^n$)**

Given $\Xi_n := \{\xi_k\}_{k=1}^p$, $\beta_n \in (0, 1)$, $\{x(r)\}_{r \in \mathbb{R}}$, and the radius $\omega$ of the covering balls. Define the new ambiguity set $\hat{\mathcal{P}}_n := \mathcal{B}_{\beta_n}(\hat{p}^n)$ where the center of the Wasserstein ball $\hat{p}^n$ is defined in (10) and the radius $\epsilon(\beta_n) := \epsilon(\beta_n) + \omega$. Then the following certificate satisfies (1) for all $x(r)$:

$$\tilde{J}_n(x(r)) := \sup_{Q \in \hat{\mathcal{P}}_n} \mathbb{E}[Q[f(x(r), \xi)], \tag{11}$$

Further, under the same assumptions required in Theorem III.2 we have the new version of $(\mathcal{P}_1^n(r))$ as follows:

$$\tilde{J}_n(x(r)) := \sup_{y_1, \ldots, y_p \in \mathbb{R}^n} \frac{1}{n} \sum_{k=1}^{p} \theta_k f(x(r), \xi_k - y_k),$$

s.t. $\frac{1}{n} \sum_{k=1}^{p} \theta_k \|y_k\| \leq \epsilon_n $.

and the associated worst-case distribution $\hat{Q}^n_n(x(r))$ is a weighted version of $Q^2_n(x(r))$ in Theorem III.2, i.e.,

$$\hat{Q}^n_n(x(r)) := \frac{1}{n} \sum_{k=1}^{p} \theta_k \delta_{\xi_k - y_k}.$$ 

where $y^* := (y_1^*, \ldots, y_p^*)$ is an optimizer of $(\mathcal{P}_1^n(r))$.

**Remark VII.1 (New version of $(\mathcal{P}_2^n(r))$)**

The equivalent formulation of Problem $(\mathcal{P}_1^n(r))$ is a new version of $(\mathcal{P}_2^n(r))$, defined as follows:

$$\tilde{J}_n(x(r)) := \max_{u_k, v_k \in \mathbb{R}^m, k \in \{1, \ldots, p\}} \frac{1}{n} \sum_{k=1}^{p} h_k^*(u_k - v_k),$$

s.t. $(u, v) \in \mathbb{R}^m$.

where for each $k \in \{1, \ldots, p\}$, $\xi_k \in \mathcal{C}_m$, and $x(r) \in \mathbb{R}^d$, we define $h_k^* : \mathbb{R}^m \rightarrow \mathbb{R}$ as

$$h_k^*(y) := \theta_k f(x(r), \xi_k - y).$$

With the constructed ambiguity set $\hat{\mathcal{P}}_n$ and certificate function $\tilde{J}_n$, the developed algorithms in Section IV and Section V are valid to solve Problem $(\hat{\mathcal{P}}_2^n(r))$. And the main Theorem VI.1 on the finite convergence of the ONLINE DATA ASSIMILATION ALGORITHM is valid for the certificate function $\tilde{J}_n$ where the only difference is that the quality of the certificate for $x_{12}^*$ in (9) is replaced by

$$\sup_{n \geq n_0} J_n^*(x_{12}^*) \leq J^* + \epsilon_1 + \epsilon_2 + \epsilon_3 + 2(1 - \frac{p_n}{n_0}) L \omega,$$

where $n_0$ is the number of the data set in $\Xi_{n_0}$ and $p_n$ indicates the number of Euclidean balls that cover $\Xi_{n_0}$.

**VIII. Simulation results**

In this section, we demonstrate the application of the ONLINE DATA ASSIMILATION ALGORITHM to find an $\epsilon_2$-optimal, $\epsilon_1$-proper data-driven decision for Problem (P) with a potentially large streaming data set.

**Case study 1 (The effect of the Incremental Covering Algorithm):**

In order to visualize the effect of the ICA, here we solve a toy problem in form of (P) using ONLINE DATA ASSIMILATION ALGORITHM, with and without ICA respectively. Let $x \in \mathbb{R}$ be the variable for Problem (P). Assume there are $N = 200$ data points $\{\xi_k\}_{k=1}^N$ streaming into the algorithm. Assume each time period is one second, and for each second $k$ we only stream in one data point $\xi_k \in \mathbb{R}^3$, where $\xi_k$ is a realization of the unknown distribution $P$. The $P$ we use for simulation is a multivariate weighted Gaussian mixture distribution with three centers, where each center has mean $\mu_1 = (2, -4, 3)$, $\mu_2 = (-3, 5, 0)$, $\mu_3 = (0, 0, -6)$, variance $\sum_1 = \text{diag}(1, 3, 2)$, $\sum_2 = 2 \cdot I_3$, $\sum_3 = I_3$, and weights $0.25, 0.5, 0.25$, respectively. Let the cost function be $f(x, \xi) := x^2 - c^T \xi$, the confidence be $1 - \beta_n := 1 - 0.95e^{-1-\sqrt{n}}$ and we use the parameter $c_1 = 2, c_2 = 1$ to design the radius $\epsilon(\beta_n)$ of the Wasserstein ball in (4). The radius of the Euclidean ball for ICA is $\omega = 1.5$. We sample the initial decision $x^{(0)}$ from the uniform distribution $[-10, 10]$. The tolerance for the algorithm is $\epsilon_1 = 10^{-5}, \epsilon_2 = 10^{-6}$.

Figure 1 and Figure 2 demonstrate the effect of ICA in ONLINE DATA ASSIMILATION ALGORITHM. Specifically, Figure 1 shows the incremental data covering at the end of the 200th time period in the $(\xi_1, \xi_2, \xi_3)$ coordinates. The large shaded blue area are 59 Euclidean balls $B_\omega$ with their centers $\{\xi_k\}_{k=1}^{59}$ denoted by the red small circles. The blue and red small circles together constitute the streamed data set $\Xi_{200} := \{\xi_k\}_{k=1}^{200}$. In Figure 2, the blue dashed line represents the number of the data points used as centers of the empirical distribution $\hat{p}^n$ over time and the black dashed line is that for distribution $P^n$. Clearly as the data streams over time, the number $p := |\mathcal{C}_n|$ is significantly smaller than $n := |\Xi_n|$, which results in the size of Problem $(\mathcal{P}_2^n(r))$ much smaller than that of $(\mathcal{P}_2^n(r))$. Further, the blue continuous line counts the total number of subproblems $(\mathcal{P}_1^n(r))$ solved for certificate generation over time and the black continuous line represents that for subproblems $(\hat{\mathcal{P}}_2^n(r))$. These subproblems search the explicit solution for the $\epsilon_1$-worst-case distribution and consume the major computing resources in ONLINE DATA.
**ASSIMILATION ALGORITHM.** It can be seen that the number of \((CP_n^{(l)})\) solved over time is on average only half of \((CB_n^{(l)})\) in each time period. Together, the dimension and total number of subproblems \((CP_n^{(l)})\) solved in the algorithm with ICA is significantly smaller than without it.

To evaluate the quality of the obtained \(\epsilon_1\)-proper data-driven decision with the streaming data, we estimate the optimizer of \((P)\), \(x^*\), by minimizing the average value of the cost function \(f\) for a validation data set of \(N_{val} = 10^4\) 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., \(J^* = J^*(x^*)\). We calculate \(J^*(x^*)\) using the underline distribution \(\mathbb{P}\), serving as the true but unknown scale to evaluate the goodness of the certificate obtained throughout the algorithm.

Figures 3 and 4 show the evolution of the certificate sequence \(\{J_n^{1}(x^{(r)})\}_{n=1}^{N_1}, \{J_n^{1}(x^{(r)})\}_{n=1}^{N_1}\) with ICA and \(\{J_n^{1}(x^{(r)})\}_{n=1}^{N_1}, \{J_n^{1}(x^{(r)})\}_{n=1}^{N_1}\) without ICA, respectively. Here, the optimal decision of \((P)\) is trivially \(x^* = 0\), and for both algorithms the subgradient counterpart of the ONLINE DATA ASSIMILATION ALGORITHM returns the optimal decision after the first data point \(\hat{\xi}_1\) is used. Therefore, after a very short period within the first second, both figures start reflecting the certificate evolution under the decision sequence \(\{x^{(r)} \approx 0\}_{r=0}^{\infty}\).

The blue line in both Figure 3 and Figure 4 show the relative goodness of the certificates for the currently used \(\epsilon_1\)-proper data-driven decision \(x^{(r)} \approx 0 \) calibrated by the estimated optimal value \(J^*\) over time. The red segments on the blue line indicate that a new certificate \(J_n^{1}(x^{(r)}(t))\) is processing when the new data set is incorporated, while at these time intervals the old certificate \(J_n^{1}(x^{(r)})\), associated with the \(\epsilon_2\)-optimal and \(\epsilon_1\)-proper data-driven decision \(x_n^{(r)}\), is still valid to guarantee the performance under the old confidence \(1 - \beta_1\). This situation commonly happens when a new data set \(\Xi_{n+1}\) is streamed in and a new certificate \(J_n^{1}(x^{(r)}(t))\) is yet to be obtained. It can be seen that after a few samples streamed, both the obtained certificate becomes close to the estimated true optimal value \(J^*\) with about the 10% range. In Figure 4 however, as the data streams over 50 seconds, the computing cost for updating certificates becomes significant for the algorithm without ICA. After 100th data point streamed, the certificate \(J_n^{1}(x^{(r)}(t))\) stops updating for all \(n \geq 100\). And, further, after all the data points streamed, the algorithm took about 70 seconds to terminate the algorithm with certificate \(J_n^{1}(x^{(r)}(t))\). This is a clear disadvantage compared to the algorithm with ICA, which terminates as soon as all the data points were taken in.

**Case study 2 (ONLINE DATA ASSIMILATION ALGORITHM with significantly large number of streaming data sets):** Here, we are to find an \(\epsilon_2\)-optimal, \(\epsilon_1\)-proper decision \(x \in \mathbb{R}^{30}\) for Problem \((P)\). We consider \(N = 500\) iid sample points \(\{\xi_e\}_{e=1}^{N}\) streaming randomly in between every 1 to 3 seconds with each data point \(\xi_e \in \mathbb{R}^{10}\) a realization of \(\mathbb{P}\). Here, we assume that the unknown distribution is a multivariate Gaussian mixture distribution with three centers where the components of the mean of each center is uniformly chosen between \([-10, 10]\), and the variance matrix is \(I_m\) for each center. We assume the cost function \(f : \mathbb{R}^{30} \times \mathbb{R}^{10} \rightarrow \mathbb{R}\) to be \(f(x, \xi) := x^\top A x + x^\top B \xi + \xi^\top C \xi\) with random values for the positive semi-definite matrix \(A \in \mathbb{R}^{30\times30}, B \in \mathbb{R}^{30\times10}\).
shows the evolution of $P$ demonstrates the incremen-
stream over time, the update of certificate to data sets streamed. Also, as the large amount of data sets
set streamed in.

Euclidean ball for ICA is
$w$ certificate becomes close to the estimated true optimal valu e
$J$ sequence
$(about
349
$\{\tilde{P}_n\}$, and the number of subproblems
$\{\tilde{C}_{P_n}^{(l)}\}_{n,l}$ solved over time periods.

and negative definite matrix $C \in \mathbb{R}^{10 \times 10}$. The radius of the Euclidean ball for ICA is $w = 5$.

Similar to Figure 2, Figure 5 demonstrates the incremental construction of the distribution $\tilde{P}^n$ and the accumulated number of Problem $\tilde{P}_n^{(r)}$ solved over time. Clearly, after certain amount of data streamed, the structure of the data set was inferred by ICA and the number of Euclidean balls used to cover the data set is about 20. Also, after 100 time period (about 100 to 200 seconds in this case), the algorithm can certify new certificate without solving any Problem $\tilde{P}_n^{(r)}$. This feature dramatically improves the performance of the Online Data Assimilation Algorithm and makes the algorithm flexible for online setting.

Similar to Figure 3, Figure 6 shows the evolution of the certificate sequence $\{J_n^{(r)}(x_n^{(r)})\}_{n=1,r=1}^{N,\infty}$ for the decision sequence $\{x_n^{(r)}\}_{n=1}^{\infty}$. Same as last case study, the obtained certificate becomes close to the estimated true optimal value $J^*$ within the 10% range after about 25 seconds with 10 data sets streamed. Also, as the large amount of data sets stream over time, the update of certificate to $J_n^{(r)}(x_n^{(r)})$ remains efficient and the algorithm terminates right after the last data set streamed in.

and
data-driven decision are approached
with a (sub)linear convergence rate. The algorithm terminates after collecting sufficient amount of data to make good decision. To facilitate the decision making, an enhanced version of the proposed algorithm is further constructed, by using an Incremental Covering Algorithm (ICA) to estimate new ambiguity sets over time. We provided sample problems and showed the actual performance of the proposed Online Data Assimilation Algorithm with ICA over time. Future work will generalize the results for weaker assumptions of the problem and potentially extend the algorithm to scenarios that include system dynamics.

IX. CONCLUSIONS

In this paper, we have proposed the Online Data Assimilation Algorithm to solve the problem in the form of (P), where the realizations of the unknown distribution (i.e., the streaming data) are collected over time in order for the real-time data-driven decision of (P) to have guaranteed out-of-sample performance. To incorporate the streaming uncertainty data, we have firstly formulated a sequence of the convex optimization problems that are equivalent to the problems for generating the certificate of the out-of-sample performance guarantee of (P), then provided a scheme that incorporates streaming data when finding the certificate for the data-driven decision and further approaching to the $\epsilon_2$-optimal and $\epsilon_1$-proper data-driven decision in real time. The data-driven decision with the certificate that guarantees out-of-sample performance are available any time during the execution of the algorithm, and the optimal data-driven decision are approached

REFERENCES

[1] D. Li and S. Martinez, “Online data assimilation in distributionally robust optimization,” in IEEE Int. Conf. on Decision and Control, Miami, FL, USA, December 2018, pp. 1961–1966.
[2] D. D. A. Shapiro et al., Lectures on Stochastic Programming: Modeling and Theory. Philadelphia, PA: SIAM, 2014, vol. 16.
[3] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, Robust optimization. Princeton University Press, 2009.
[4] 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.
[5] R. Gao and A. Kleywegt, “Distributionally robust stochastic optimization with Wasserstein distance,” arXiv preprint arXiv:1604.02199, 2016.
III.1. We see that each column vector of the matrix $A_n := \left[ \begin{array}{c} \mathcal{E} \end{array} \right]_n$ is a concatenation of an extreme point of Problem (P1(r)). Let us denote the matrix $A_n = \left[ \begin{array}{c} \mathcal{E} \end{array} \right]_n \in \mathbb{R}^{m \times 2mn}$ by $\mathcal{E}$. By construction of Problem (P2(r)), we see that each column vector of the matrix $\mathcal{E} \in \mathbb{R}^{m \times 2mn}$ is included. Then, any feasible solution of (P1(r)) can be written as $\tilde{y} = \mathcal{E} \mathcal{E} \mathcal{E} \in \mathbb{R}^{m \times 2mn}$, where $\mathcal{E}$ is a vector of the convex combination coefficients of the extreme points of the constraint set in (P1(r)). Clearly, we have $(\tilde{u}, \tilde{v}) \in \Delta_{2mn}$, i.e., $\mathcal{E} \in \mathbb{R}^{m \times 2mn}$. Then, by construction $(\tilde{u}, \tilde{v}) := \mathcal{E}$ is feasible for (P2(r)).

For 3, since Problem (P1(r)) and (P2(r)) are the same in the sense of (1) and (2), then if $\mathcal{E}$ is an optimizer of (P2(r)), by letting $\mathcal{E}$ the global linear convergence of Frank-Wolfe optimization variants,” in Advances in Neural Information Processing Systems, 2015, pp. 496–504.

IV.2

L. V. Kantorovich and G. S. Rubinstein, “On a space of completely additive functions,” Vestnik Leningrad. Univ. 13, no. 7, pp. 52–59, 1958.

15. S. M. Robinson, “Linear convergence of epsilon-subgradient descent methods for a class of convex functions,” Mathematical Programming, vol. 86, no. 1, pp. 41–50, 1999.

16. M. P. T. Larsson and A. Strömberg, “On the convergence of conditional epsilon-subgradient methods for convex programs and convex–concave saddle-point problems,” European Journal of Operational Research, vol. 151, no. 3, pp. 461–473, 2003.

17. A. Nedić and D. P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” SIAM Journal on Optimization, vol. 12, no. 1, pp. 109–138, 2001.

18. A. Nedić and D. Bertsekas, “Convergence rate of incremental subgradient algorithms,” in Stochastic optimization: algorithms and applications, 2001, pp. 223–264.

19. D. P. Bertsekas, A. Nedić, and A. Ozdaglar, Convex analysis and optimization. Athena Scientific, 2003.

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

21. D. Li and S. Martínez, “Online data assimilation in distributionally robust optimization,” arXiv preprint arXiv:1803.07984, 2018.

22. 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.

IV.1

Proof: Given tolerance $\epsilon_1$, decision $\sigma(r)$ and data set $\Xi_n$, let $H_n : \mathbb{R}^{mn} \to \mathbb{R}$, $H_n := \frac{1}{n} \sum_{k=1}^{n} h_k^r$ denote the objective function of (P2(r)) and let $S_n$ denote the family of subsets of $\Delta_{2mn}$. In the procedure of Algorithm 1, let us consider a sequence of generated candidate vertex sets: $I_n \subset I_n^{(r)}$, $l = 0, 1, 2, \ldots$ with $I_n(0) \subseteq S_n$. We show the convergence of Algorithm 1 for any data set $\Xi_n$, by two steps.

Step 1) The sequence of $I_n(0) \subseteq I_n^{(r)}$ is finite and the number of iterations is at most $2mn$. For each $l$ and candidate optimizer $\mathcal{E}$, we generate a nonempty set of search points $\Omega(l)$ with suboptimality gap $\eta(l)$ via (LP(l)) where $\eta(l) \leq \epsilon_1$, then we solved (P2(r)) to $\epsilon_1$-optimality and $l$ is therefore finite, otherwise we update $I_n(l) := I_n^{(l)} \cup \Omega(l)$, given that the maximal cardinality of each $I_n(l) \in S_n$ is bounded by $2mn$, then it is sufficient to show $\Omega(l) \cap I_n^{(l)} = \emptyset$. By $\eta(l)$ is an $\epsilon_1$-optimal of (CP(l)) under convex $\Omega(l)$, then for any $y \in \Omega(l)$, it holds that $\frac{1}{n} \sum_{k=1}^{n} \left( \nabla h_k^r(y_k - y_k(l-1)) \right) \leq \epsilon_1$. Since any element in $\Omega(l)$ is such that $\eta(l) > \epsilon_1$, then for any $y \in \Omega(l) \cap I_n(l)$, we have $\mathcal{E} \neq \Omega(l)$, which concludes $\Omega(l) \cap I_n^{(l)} = \emptyset$. Further, the cardinality of $\Omega(l)$ is at least one for every iteration $l$, then after at most $2mn$ steps the cardinality of $I_n(l)$ becomes $2mn$, which implies the $\epsilon_1$-optimality of (P2(r)) by the $\epsilon_1$-optimality of (CP(l))

Step 2) The computational bound of Algorithm 1 is quantified. To see this, consider the problems (LP(l)) and ((CP(l))) of $\mathcal{E}$.
the gradients, the computation of \((\text{LP}^{(l)})\) is negligible. Thus, the
computational bound is given by the sum of the steps to solve the \(\{(\text{CP}^{(l)})\}l\), where the number of iterations \(l = 2mn\) in
the worst case. For each \(\text{CP}^{(l)}\) solved by AFWA, index the
AFWA iterations by \(i = 0, 1, 2, \ldots\), let \(\text{obj}^{(l)}(i)\) be the objective
value at each iteration, and assume the optimal objective value is \(\text{obj}^{(l)}(0)\). As then in Theorem II.1, let \(\kappa_l \in (0,1) \subset \mathbb{R}\) be the
decay parameter related to local strong concavity of \(H_n\) over
\(\text{conv}(I_n^{(l)})\). Then using the linear convergence rate of the
AFWA, each \(\text{CP}^{(l)}\) achieves the following computational bound:
\[
\text{obj}^{(l)}_i - \text{obj}^{(l)}_0 \leq \kappa_l^i(\text{obj}^{(l)}_0 - \text{obj}^{(l)}_0),
\]
where the initial condition \(\text{obj}^{(0)}_0\) results from an \(\epsilon_1\)-optimal
optimizer of CP at iteration \(l - 1\), i.e., we can equivalently
denote \(\text{obj}^{(l)}_0\) by \(J^{(l-1)}(x^{(r)})\), for all \(l \in \{1, \ldots, 2mn\}\).

Let us consider sequence \(\{(\text{CP}^{(l)})\}_l\) with feasible sets
\(\text{conv}(I^{(l)}_n)\). Then we have
\[
\text{conv}(I^{(0)}_n) \subset \text{conv}(I^{(1)}_n) \subset \cdots \subset \text{conv}(I^{(2mn)}_n).
\]
This results into monotonically decaying and \((\epsilon_1\text{-})\)optimal
objective values, as given in the following
\[
0 < \kappa_1 \leq \kappa_2 \leq \cdots \leq \kappa_{2mn} < 1,
\]
\[
J^{(0)}_n(x^{(r)}) \leq J^{(1)}_n(x^{(r)}) \leq \cdots \leq J^{(2mn)}_n(x^{(r)}),
\]
\[
\text{obj}^{(0)}_0 \leq \text{obj}^{(1)}_0 \leq \cdots \leq \text{obj}^{(2mn)}_0.
\]
Using previous notation, we can identify \(J^{(2mn)}_n(x^{(r)}) =
J^{(n)}_n(x^{(r)}), \text{obj}^{(0)}_0 = J^{(0)}_n(x^{(r)}), \text{and \ obj}^{(2mn)}_0 =
J^{(n)}_n(x^{(r)}). \) Let us denote \(\kappa := \max\{\kappa_l\}\). Then, by solving each \(\text{CP}^{(l)}\) to
\(\epsilon_1\)-optimality, it leads to the accumulated computational steps
\(\phi(n) := \sum_i \kappa_i\), where each \(\kappa_i\) is the computation step for \(\epsilon_1\)-
optimal \(\text{CP}^{(l)}\) that satisfies the following inequality:
\[
\kappa_i^i(J_n(x^{(r)}) - J^{(0)}_n(x^{(r)})) \leq \epsilon_1, \quad l \in \{1, \ldots, 2mn\}.
\]

Finally, in the worst-case scenario, the computational bound of
Algorithm 1 is
\[
\phi(n) \leq (2mn)\log_\frac{\epsilon_1}{J_n(x^{(r)}) - J^{(0)}_n(x^{(r)})}.
\]
Next, we show the convergence of Algorithm 1 under online data
sets \(\{\Xi_n\}_{n=1}^N\). Similarly to the proof for the computational
bound for a given \(n\), we can compute the worst-case bound under
\(\{\Xi_n\}_{n=1}^N\), by summing over the steps required
to solve the \(\{(\text{CP}^{(l)})\}_n.l\). This leads to the stated bound \(\phi(n)\),
where the empirical cost \(J^{(n)}_N(x^{(r)})\) serves as the cost of
initial condition \(y^{(0)} := 0_{2mn}\). In this way, when the average
data-streaming rate is slower than \((\phi(1))^{-1}\), we claim that
Algorithm 1 always can find the certificate for each data set
\(\Xi_n\). This is because in each time period \(n\) on average, we only have
\(2mn\) extreme points, and \(2mn(n-1)\) has been explored due to the adaptation of the candidate vertex set \(I_0^{(n)}\).

**Lemma V.1**

**Proof:** For any \(x, y \in \mathbb{R}^d\) and \(t \in [0,1] \subset \mathbb{R}\), we have
the point \(z = tx + (1-t)y \in \mathbb{R}^d\) and an optimizer of (5),
\[
\text{Q}_n^*(z), \text{ such that }
J_n(z) \leq \mathbb{E}\text{Q}_n^*(z) [f(x, \xi) + (1-t)f(y, \xi)]
= \mathbb{E}\text{Q}_n^*(z) [f(x, \xi)] + (1-t)\mathbb{E}\text{Q}_n^*(z) [f(y, \xi)]
\leq tJ_n(x) + (1-t)J_n(y).
\]

**Lemma V.2**

**Proof:** Let us consider the function \(\mathbb{E}\text{Q}_n^*(x|z) [f(x, \xi)]\).
Using Assumption V.1 on convexity of \(f\) in \(x\), we have for any \(z \in \text{dom} J_n\) the following relation:
\[
\mathbb{E}\text{Q}_n^*(x|z) [f(x, \xi)] \geq J_n^*(x|z) + g_n^*(x|z) \cdot (z - x|z).
\]
Knowing that \(J_n^*(x|z) \geq J_n(x|z) - \epsilon_1\) and \(J_n(z) \leq
\mathbb{E}\text{Q}_n^*(x|z) [f(x, \xi)],\) this concludes the first part of the proof.

To show the second part, similarly, we also have for any \(x, z \in \text{dom} J_n\) the following relation:
\[
\mathbb{E}\text{Q}_n^*(x|z) [f(x, \xi)] \geq \mathbb{E}\text{Q}_n^*(x|z) [f(\tilde{x}, \xi)] + g_n^*(\tilde{x}|z) \cdot (z - \tilde{x} |z).
\]
Using Point Search Algorithm 2, we achieve an \(\eta > 0\) such that
\(\mathbb{E}\text{Q}_n^*(x|z) [f(\tilde{x}, \xi)] \geq J_n(\tilde{x}) - \eta\). Finally, by similar
statement as in the first part, we claim \(g_n^*(\tilde{x}) \in \partial \phi J_n(\tilde{x}).\)

**Lemma V.3**

**Proof:** In the \(n^{th}\) time period, let us consider subgradient
iterates \(i \) for all \(r_n \leq i \leq r\):
\[
\|x^{(i+1)} - x_n^*\|^2 = \|x^{(i)} - x_n^* - \alpha^{(i)} \cdot g_n^{(i)}(x^{(i)})\|^2
\leq \|x^{(i)} - x_n^*\|^2 + (\alpha^{(i)})^2 \min\{\|g_n^{(i)}(x^{(i)})\|^2, 1\}
\leq \|x^{(i)} - x_n^*\|^2 + 2(\alpha^{(i)})^2 J_n(x_n^*) - J_n(x^{(i)}) + \epsilon_{SA}\),
\]
From Lemma V.2, we know that \(J_n(x_n^*) \geq J_n(x^{(i)}) +
g_n^{(i)}(x_n^*) \cdot (x_n^* - x^{(i)}) - \epsilon_{SA}\) for all \(x_n^*\). Then, we have
\[
\|x^{(i+1)} - x_n^*\|^2 \leq \epsilon_{SA}^2 \min\{\|g_n^{(i)}(x^{(i)})\|^2, 1\}
+ \|x^{(i)} - x_n^*\|^2 + 2(\alpha^{(i)})^2 J_n(x_n^*) - J_n(x^{(i)}) + \epsilon_{SA}\).
\]
Combining the inequalities over iterations from \(r_n \) to \(r\),
\[
0 \leq \|x^{(r_n)} - x_n^*\|^2 + \sum_{i=r_n}^{r} (\alpha^{(i)})^2 \min\{\|g_n^{(i)}(x^{(i)})\|^2, 1\}
+ \sum_{i=r_n}^{r} 2(\alpha^{(i)})^2 J_n(x_n^*) - J_n(x^{(i)}) + \epsilon_{SA}\)
\leq \|x^{(r_n)} - x_n^*\|^2 + 2(1 - m) \sum_{i=r_n}^{r} (\alpha^{(i)})^2
+ \sum_{i=r_n}^{r} 2(\alpha^{(i)})^2 J_n(x_n^*) - J_n(x^{(i)}) + \epsilon_{SA}\).
\]
Then, using the fact that
\[
\sum_{i=r_n}^{r} 2\alpha(i) (J_n(x_n^i) - J_n(x_i)) \leq \min_{k\in\{r_n,\ldots,r\}} \{ J_n(x(k)) - J_n(x_n^i) \} \leq \frac{\mu}{r} \sum_{i=r_n}^{r} (\alpha(i))^2 + \mu \epsilon_{SA},
\]
and the previous iteration, we have
\[
\min_{k\in\{r_n,\ldots,r\}} \{ J_n(x(k)) - J_n(x_n^i) \} \leq \frac{\mu}{r} \sum_{i=r_n}^{r} (\alpha(i))^2 + \mu \epsilon_{SA}.
\]
Since we have \(\sum_{i=r_n}^{r} \alpha(i) = \infty, \sum_{i=r_n}^{r} (\alpha(i))^2 < \infty,\) and as \(r\) increases to \(\infty,\) we have the right hand side term goes to \(\mu \epsilon_{SA} < \epsilon_2,\) then there exists a large enough but finite number \(\bar{r},\) such that the right hand side of the above inequality is no greater than \(\epsilon_2,\) which concludes the claim.

**THEOREM V.1**

**Proof:** The computational bound to achieve an \(x_n^{\epsilon_2}\) strongly depends on the subgradient iterations \(\bar{r} := r_{n+1} - r_n\) in Lemma V.3 and the number of subgradient functions \(g_n^{r_{n+1}}\) constructed via Algorithm 1. To characterize this bound, we quantify the computational steps for \(g_n^{r_{n+1}}\) next.

For each time period \(n,\) let us assume the Certificate Generation Algorithm 1 has explored the feasible set of \((P2)_{r}\) when obtaining the initial certificate \(J_{n}^{1}(x^{r_n})\). This procedure consumes a worst-case computational time \(\phi(n),\) (or \(\bar{\phi}(1)\) if a data-streaming scenario), as stated in Theorem IV.1. After this initial step, every time the Subgradient Algorithm needs to execute Algorithm 1 at some \(r \leq r_{n+1}.\) Algorithm 1 will solve a unique \((CP_{r})\) and return \(Q_{r}\) for an \(\epsilon_1\)-subgradient function \(g_n^{r}\) at \(x^{r}\). Let \(CP_{r}\) denote the unique \((CP_{r})\) solved at \(x^{r}\). Then, to quantify the computational steps for \(g_n^{r_{n+1}}\) we compute the sum of the steps to solve \(CP_{r}\).

Let us denote the number of steps solving \(CP_{r}\) by \(i_r,\) for all \(r \in \{r_{n+1},\ldots,r_n\}\). Then, we aim to quantify \(i_{r+1}\) for \(g_n^{r+1}\). To achieve this, let us assume a subgradient function \(g_n^{r}\) is computed at an iteration \(r.\) Then we perform a subgradient iteration (7) and obtain an \(x^{r+1}\). By using a subgradient estimation strategy, we obtain the optimality gap \(\eta^{(1)}\) via Algorithm 2, denoted by \(\eta_r^{(1)} := \eta^{(1)}\). This gap \(\eta^{(1)}\) enables us to quantify the distance between the initial objective value and the optimal objective value of \(CP_{r+1}\). When \(\eta^{(1)} \leq \epsilon_{SA}\), the algorithm uses the estimated subgradient function and \(i_{r+1} = 0.\) Otherwise, the computational steps can be calculated via convergence of AFWA for \(CP_{r+1}\) by
\[
\kappa^{i_{r+1}} \eta_r^{(1)} \leq \epsilon_1,
\]
where \(\kappa\), or using \(\bar{\kappa}\) for the data-streaming case, is determined as in Theorem IV.1. Let us consider a threshold value \(t_r\) as the following:
\[
t_r := \begin{cases} \epsilon_1, & \text{if } \eta_r \leq \epsilon_{SA}, \\ \eta_r, & \text{o.w.} \end{cases}
\]
Then we can represent each value \(i_r\) by
\[
i_r = \log_{\eta_r}(\frac{\epsilon_1}{t_r}), \ r \in \{r_{n},\ldots,r_{n+1}\}.
\]
Let us denote \(t := \max_r(t_r)\). Then, the computational steps for \(g_n^{r_{n+1}}\) are bounded by
\[
\sum_{r} i_r \leq \bar{r}\log_{\eta_r}(\frac{\epsilon_1}{t_r}).
\]
Finally, the computational steps to achieve an \(x_n^{\epsilon_2}\), denoted by \(\phi(n, \bar{r}) := \phi(n) + \sum_r i_r + \bar{r},\) are bounded as:
\[
\varphi(n, \bar{r}) \leq \phi(n) + \bar{r} \left( \log_{\eta_r}(\frac{\epsilon_1}{t_r}) + 1 \right).
\]
Again, one should use \(\bar{\phi}(1)\) in the bound in place of \(\phi(n)\) if considering a data-streaming scenario.

**THEOREM VI.1**

**Proof:** The first part of the proof is an application of Theorem IV.1 and Lemma V.3. For any data set \(\Xi_n\) and the initial data-driven decision \(x^{r_n}\), by Theorem IV.1 we can show \(x^{r_n}\) to be \(\epsilon_1\)-proper, via finding \(J_n^{1}(x^{r_n})\) such that \(P_n^{n}([E_{\Xi_n}[x^{r_n}], \xi]) \leq J_n^{1}(x^{r_n}) + \epsilon_1 \geq 1 - \beta_n.\) Then using Lemma V.3, an \(\epsilon_2\)-optimal \(\epsilon_1\)-proper data-driven decision \(x_n^{\epsilon_2}\) with certificate \(J_n^{1}(x_n^{\epsilon_2})\) can be achieved. Therefore the performance guarantee (2) holds for \(x_n^{\epsilon_2}\), i.e., \(P_n^{n}([E_{\Xi_n}[x_n^{\epsilon_2}], \xi]) \leq J_n^{1}(x_n^{\epsilon_2}) + \epsilon_1 \geq 1 - \beta_n.\)

Then we show the almost sure performance guarantee. For any time period \(n,\) the algorithm guarantees to find \(x_n^{\epsilon_2}\) with the performance guarantee (2), which can be equivalently written as \(P_n^{n}([E_{\Xi_n}[x_n^{\epsilon_2}], \xi]) \geq J_n^{1}(x_n^{\epsilon_2}) + \epsilon_1 \geq 1 - \beta_n.\) As \(\sum_{n=1}^{\infty} \beta_n < \infty,\) from the 1st Borel-Cantelli Lemma we have that \(P_n^{\infty}([E_{\Xi_n}[x_n^{\epsilon_2}], \xi]) \geq J_n^{1}(x_n^{\epsilon_2}) + \epsilon_1 \) occurs infinitely many often = 0. That is, almost surely we have that \(E_{\Xi_n}[x_n^{\epsilon_2}], \xi) \geq J_n^{1}(x_n^{\epsilon_2}) + \epsilon_1 \) occurs at most for finite number of \(n.\) Thus, there exists a sufficiently large \(n_0,\) such that for all \(n \geq n_0,\) we have \(E_{\Xi_n}[x_n^{\epsilon_2}], \xi) \leq J_n^{1}(x_n^{\epsilon_2}) + \epsilon_1 \) occurs almost surely, i.e., \(P_n^{n}([E_{\Xi_n}[x_n^{\epsilon_2}], \xi]) \leq J_n^{1}(x_n^{\epsilon_2}) + \epsilon_1 \) = 1 for all \(n \geq n_0.\) Later if we pick \(n_0 \geq n_1,\) then the almost sure performance guarantee holds for such \(x_n^{\epsilon_2}\) and \(J_n^{1}(x_n^{\epsilon_2}).\)

Now, it remains to find an \(n_0,\) associated with an \(\epsilon_2\)-optimal and \(\epsilon_1\)-proper data-driven decision \(x_n^{\epsilon_2},\) such that the performance bound (9) of the certificate \(J_n^{1}(x_n^{\epsilon_2})\) can be guaranteed for the termination of the ONLINE DATA ASSIMILATION ALGORITHM.

First, let \(x^{\delta}\) denote the \(\delta\)-optimal solution of (P), i.e., \(E_{\Xi_n}[x^{\delta}], \xi) \leq J^* + \delta.\) By construction of the certificate in the algorithm we have \(J_n^{1}(x_n^{\epsilon_2}) \leq J_n(x_n^{\epsilon_2}) \leq J_n(x_n^{\epsilon_2}) + \epsilon_2 \leq J_n(x^{\delta}) + \epsilon_2 \leq J_n^{1}(x^{\delta}) + \epsilon_1 + \epsilon_2 \) for all \(n,\) where the first inequality holds because \(J_n^{1}\) is the function that achieves the supreme of Problem (5) while \(J_n^{1}(x_n^{\epsilon_2})\) is the
objective value for a feasible distribution $\mathbb{Q}_n^{\alpha}(x^o)$, the second inequality holds because $x^*_{\alpha}$ is $\epsilon_2$-optimal, the third inequality holds because $x^*_{\alpha}$ is a minimizer of the certificate function $J^\alpha_n$, the last inequality holds because the Algorithm 1 for certificate generation guarantees the existence of $J^\alpha_n(x^\alpha)$ such that $J^\alpha_n(x^\alpha) \leq J^\alpha_n(x^\alpha) + \epsilon_1$, with an distribution $\mathbb{Q}_n^{\alpha}(x^\alpha)$ satisfying $d_W(\mathbb{P}^n, \mathbb{Q}_n^{\alpha}(x^\alpha)) \leq \epsilon(\beta_n)$.

Next, we exploit the connection between $J^\alpha_n(x^\alpha)$ and $J^\star$. By Assumption IV.1 on the concavity of $f$ in $\xi$, there exists a constant $L > 0$ such that $f(x, \xi) \leq L(1 + ||\xi||)$ holds for all $x \in \mathbb{R}^d$ and $\xi \in \mathbb{Z}$. Then by the dual representation of the Wasserstein metric from Kantorovich and Rubinstein [4, 14] we have $J^\alpha_n(x^\alpha) := \mathbb{E}[\mathbb{Q}_n^{\alpha}(x^\alpha)] f(x^\alpha, \xi)] \leq \mathbb{E}[f(x^\alpha, \xi)] + Ld_W(\mathbb{P}, \mathbb{Q}_n^{\alpha}(x^\alpha))$. In order to quantify the last term, we apply the triangle inequality, which gives us $d_W(\mathbb{P}, \mathbb{Q}_n^{\alpha}(x^\alpha)) \leq d_W(\mathbb{P}, \mathbb{P}^n) + d_W(\mathbb{P}^n, \mathbb{Q}_n^{\alpha}(x^\alpha))$. Then by the performance guarantee we have $\mathbb{P}^n \{d_W(\mathbb{P}, \mathbb{P}^n) \leq \epsilon(\beta_n)\} \geq 1 - \beta_n$, and by the the way of constructing $\mathbb{Q}_n^{\alpha}(x^\alpha)$ we have $d_W(\mathbb{P}^n, \mathbb{Q}_n^{\alpha}(x^\alpha)) \leq \epsilon(\beta_n)$. These inequalities result in $\mathbb{P}^n \{d_W(\mathbb{P}, \mathbb{Q}_n^{\alpha}(x^\alpha)) \leq 2\epsilon(\beta_n)\} \geq 1 - \beta_n$, or equivalently, $\mathbb{P}^n \{d_W(\mathbb{P}, \mathbb{Q}_n^{\alpha}(x^\alpha)) \geq 2\epsilon(\beta_n)\} \leq \beta_n$. As $\sum_{n=1}^{\infty} \beta_n < \infty$, then the 1st Borel-Cantelli Lemma applies to this situation. Thus we claim that there exists a sufficiently large $n_3$ such that for all $n \geq n_3$ we have $\mathbb{P}^n \{d_W(\mathbb{P}, \mathbb{Q}_n^{\alpha}(x^\alpha)) \leq 2\epsilon(\beta_n)\} = 1$. We use this bound to deal with the last term in the upper bound of $J^\alpha_n(x^\alpha)$. In particular, we have $\mathbb{P}^n \{J^\alpha_n(x^\alpha) \leq \mathbb{E}[f(x^\alpha, \xi)] + 2L\epsilon(\beta_n)\} = 1$ for all $n \geq n_2$. As $\epsilon(\beta_n)$ decreases and goes to 0 as $n \to \infty$, there exists $n_3$ such that $2L\epsilon(\beta_n) \leq \epsilon$ holds for all $n \geq n_3$. Therefore, we have $\mathbb{P}^n \{J^\alpha_n(x^\alpha) \leq \mathbb{E}[f(x^\alpha, \xi)] + \epsilon\} = 1$ for all $n \geq \max\{n_2, n_3\}$.

Combining all the inequalities of the above results, we obtain almost surely $J^\alpha_n(x^\alpha_{\alpha}) \leq J^\star + \delta + \epsilon_1 + \epsilon_2 + \epsilon_3$, for all $n \geq \max\{n_2, n_3\}$. Since $\delta$ can be arbitrarily small, then by letting $\eta_0 := \max\{n_1, n_2, n_3\}$ we have almost sure performance guarantee $\mathbb{P}^{\eta_0} \{\mathbb{E}[f(x^\alpha_{\alpha}, \xi)] \leq J^\alpha_n(x^\alpha_{\alpha}) + \epsilon_1\} = 1$, and almost surely
\[
\sup_{n \geq \eta_0} J^\alpha_n(x^\alpha_{\alpha}) \leq J^\star + \epsilon_1 + \epsilon_2 + \epsilon_3.
\]

**Lemma VII.1**

**Proof:** The proof is an application of the dual characterization of the Wasserstein distance. Let us consider
\[
d_W(\mathbb{P}^n, \mathbb{P}^n) = \sup_{f \in \mathcal{L}} \left\{ \int_{\mathbb{R}^d} f(\xi) \mathbb{P}^n(d\xi) - \int_{\mathbb{R}^d} f(\xi) \mathbb{P}^n(d\xi) \right\},
\]
\[
= \frac{1}{n} \sup_{f \in \mathcal{L}} \left\{ \sum_{k=1}^{n} f(\xi_k) - \sum_{k=1}^{p} \theta_k f(\xi_k) \right\}.
\]
By partitioning the data set $\Xi_n$ into $C_n$ and $\Xi_n \setminus C_n$ for each summation term, we have
\[
\sum_{k=1}^{n} f(\xi_k) = \sum_{\xi \in C_n} f(\xi) + \sum_{\xi \in \Xi_n \setminus C_n} f(\xi),
\]
\[
\sum_{k=1}^{p} \theta_k f(\xi_k) = \sum_{k=1}^{p} f(\xi_k) + \sum_{\xi \in \Xi_n \setminus C_n} \ell^{-1} \sum_{k \in \ell} f(\xi_k).
\]
Canceling the first summation term gives us the following
\[
d_W(\mathbb{P}^n, \mathbb{P}^n) = \frac{1}{n} \sup_{f \in \mathcal{L}} \left\{ \sum_{\xi \in C_n} f(\xi) - \sum_{\xi \in \Xi_n \setminus C_n} f(\xi) \right\},
\]
\[
\leq \frac{1}{n} \sup_{f \in \mathcal{L}} \left\{ \sum_{\xi \in C_n} f(\xi) - \sum_{k \in \ell} f(\xi_k) \right\},
\]
\[
\leq \frac{1}{n} \sum_{\xi \in \Xi_n \setminus C_n} \ell^{-1} \sum_{k \in \ell} f(\xi_k),
\]
where the first inequality is derived by taking component-wise absolute value; the second inequality comes from the fact that $f$ is in the space of all Lipschitz functions defined on $\Xi$ with Lipschitz constant 1; and the third inequality is because $\xi \in B_{\omega}(\xi_k)$.

**Lemma VII.2**

**Proof:** From Lemma III.1 we have $\mathbb{P}^n \{d_W(\mathbb{P}, \mathbb{P}^n) \leq \epsilon(\beta_n)\} \geq 1 - \beta_n$ for each $n$. Then using the result from Lemma VII.1 we have $\mathbb{P}^n \{d_W(\mathbb{P}, \mathbb{P}^n) \leq d_W(\mathbb{P}^n, \mathbb{P}^n) + d_W(\mathbb{P}, \mathbb{P}^n) \leq \epsilon(\beta_n) + \omega\} \geq 1 - \beta_n$, i.e., $\mathbb{P}^n \{d_W(\mathbb{P}, \mathbb{P}^n) \leq \epsilon(\beta_n)\} \geq 1 - \beta_n$ for each $n$. The rest of the proof follows directly from that in Lemma III.1 and Theorem III.2.