On the complexity of the data-driven wasserstein distributionally robust binary problem

Hyoseok Kim, Dimitri Watel, Alain Faye, Hervet Cédric

On the complexity of the data-driven Wasserstein distributionally robust binary problem

Hyoseok Kim (CEDRIC-CNAM, Kardinal)
Dimitri Watel (SAMOVAR, ENSIIE)
Alain Faye (CEDRIC-CNAM, ENSIIE)
Cédric Hervet (Kardinal)

Abstract

In this paper, we use a data-driven Wasserstein distributionally robust framework to consider uncertain parameters in optimization problems.

Distributionally robust optimization is an approach to optimization under uncertainty that assumes only partial information on the probability distribution of the uncertain parameters. Unlike the classic approach of stochastic optimization, in DRO, the exact probability distribution is unknown. Instead, we assume that it belongs to an ambiguity set of distributions. The ambiguity set we use is a Wasserstein ball which is, using the Wasserstein metric, the ball centered at the empirical distribution of the training samples dataset, with a chosen radius ε. This particular case of DRO is called data-driven Wasserstein DRO.

In the case of a combinatorial optimization problem when only the cost function is affected by uncertainties, we show that the data-driven Wasserstein distributionally robust counterpart of a polynomial problem remains polynomial. More precisely, we prove that, if the optimization problem can be written as a 0-1 integer linear program with n variables, the complexity of solving the distributionally robust counterpart is at most $n + 1$ times the complexity of solving the original problem. This means that every complexity results (related to polynomiality) of an optimization problem is kept for its robust counterpart. For example, the robust counterpart of any α-approximable NP-hard 0-1 discrete problem is also α-approximable.

Keywords: distributionally robust optimization, data-driven wasserstein distributionally robust optimization, optimization under uncertainty, computational complexity, approximability

1 Introduction

Stochastic optimization and robust optimization are two classic approaches to consider uncertain parameters in optimization problems. In stochastic optimization, the probability distribution of uncertainties is assumed to be known. Usually, uncertainty affects the objective function and the goal is to compute a feasible solution minimizing the expected cost. In robust optimization, uncertainty is described by a set of scenarios, each scenario setting the objective function, and the goal is to compute a solution minimizing the worst case objective function among all the scenarios. Distributionally robust optimization (DRO) is another modeling paradigm that can be seen as the unification of the previous two approaches [4, 9]. We are given a set of distributions of the uncertainties and the goal is to compute a feasible solution minimizing the expected cost of the worst case distribution. Data-driven Wasserstein DRO (WDRO) is a particular case of DRO where a central distribution P is estimated given a set of historical data and the set of distributions is defined, given a real $\varepsilon > 0$ as the ball of radius ε centered at P using the p-Wasserstein metric for some given $p \in \overline{\mathbb{R}}^{+}$.

Exemple 1. A delivery man must choose a path among three. Each day the traffic is subject to uncertainty and affects the driving time of the delivery. A set of 4 historical data was measured. Based on this data, the delivery man must choose the path he will follow tomorrow.

paths	S_1	S_2	S_3	S_4	Tomorrow
P_1	1	1	8	6	?
P_2	6	6	5	3	?
P_3	7	3	8	2	?

In stochastic optimization, the distribution is usually known. In this example, we assume that the distribution can be deduced from the data (known as sample average approximation), for instance, by assuming each of the four scenarios may appear with the same probability tomorrow: in that case the expected value of P_1, P_2 and P_3 are respectively 4, 5 and 5, leading to the decision P_1. In robust optimization, we assume one of the four scenarios will appear tomorrow and we want to choose the path that minimizes the worst case scenario. For P_1, P_2 and P_3, this worst case scenario is respectively S_3 with value 8, S_1 with value 6 and S_3 with value 8, leading to the decision P_2.

DRO is an extension of stochastic optimization and robust optimization where the real distribution of the data is unknown. For instance, we assume the delivery man knows that the real distribution for tomorrow is one of the two possible distributions:

- Distribution D_1 : each of S_1 and S_2 may appear with a probability $\frac{1}{2}$
- Distribution D_2 : each of S_3 and S_4 may appear with a probability $\frac{1}{2}$

With no more information on which one is the real distribution, a DRO decision consists in computing the expected value of each path for each distribution and then choosing the path for which the worst expected value is minimum.

With D_1, the expected values of P_1, P_2 and P_3 are respectively 1, 6 and 5. With D_2, the expected values are respectively 7, 4 and 5. If the delivery man

chooses P_1, the worst case distribution is D_2, with an expected value of 7. If he chooses P_2, D_1 is the worst with an expected value of 6. If he finally chooses P_3, D_2 is the worst with an expected value of 5. This finally leads to the choice of P_3.

Usually, computing a list of candidates distributions is not possible. This is why we use particular cases of DRO like WDRO. In this case, we consider that the estimated distribution D used in the stochastic optimization model (each scenario may appear with a probability $\frac{1}{4}$) is a good approximation of the real distribution but not the real one which is included in a ball around D. A formal definition of WDRO is given in Section 2.

DRO can be seen as the unification of stochastic and robust optimization. In practice, the decision maker does not know the real distribution. Thus, DRO offers more robustness than stochastic optimization because instead of considering that we know the real distribution, we optimize inside of a set of candidate distributions. In general, DRO is less conservative than optimizing the worst-case scenario in robust optimization because an historical data can contain outliers due to errors or bad measures. When the distribution set is reduced to one single distribution D, DRO is equivalent to stochastic optimization using D as the real distribution.

Recently, data-driven Wasserstein DRO gained attention in operations research and machine learning literature [5, 6, 8]. As a generalization of the stochastic optimization and the robust optimization, WDRO is obviously NP-Hard. In [5], the authors prove different results on WDRO and especially, for some particular objective functions, computing the worst case distribution given a feasible solution is a tractable problem.

In [1], the authors give a framework to transform a combinatorial problem into a particular robust optimization problem and show that the transformation preserves the polynomial complexity and approximability when only the objective function is subject to uncertainty. Similarly, the data-driven Wasserstein DRO can be seen as a framework to transform a combinatorial problem into a DRO problem. In this paper, we show that this transformation also preserves the polynomial complexity and approximability of the problem when only the objective function is subject to uncertainty, whatever the parameter p chosen for the p-Wasserstein distance. This implies, for instance, the WDRO-Shortest Path problem, the WDRO-minimum spanning tree problem and the WDRO-Knapsack (with polynomial weights) problem remain polynomial. In addition, polynomial approximability of NP-Hard problems like the WDRO-Steiner tree problem, the WDRO-Traveling Salesman problem and the WDRO-Set Cover problem are preserved.

To do so, we consider the general case where the combinatorial optimization problem is written as a 0-1 integer linear program and is given with a black box exact algorithm or a black box approximation algorithm with ratio *alpha* for the problem. Our framework transforms this program into an instance of the WDRO counterpart of the problem. We show, by reformulating this instance into a deterministic mathematical program, that it can be solved or approximated to within the same ratio α using at most $n + 1$ times the black box algorithm.

Note that we do not consider uncertainties on the constraints coefficients, as each scenario would lead to a distinct set of feasible solutions and it would

not be possible to define a unique solution satisfying all the constraints for all the scenarios. A way to deal with those uncertainties is, for instance, the distributionallly robust chance constrained model [3, 7, 10].

In the second section, we first define the necessary notations and concepts used in this paper (as the Distributionally Robust Optimization and the p-Wasserstein distance), and secondly we define formally our framework. We then, in the third section, reformulate the problem into a deterministic mathematical program. The fourth section is dedicated to describing our main algorithm and proving our complexity result.

2 Definition of the framework

This section is dedicated to giving the necessary notations and concepts used in this paper. We first review models for optimizing under uncertainty, including the Distributionally Robust Optimization and the data-driven Wasserstein DRO. We finish by explaining how we transform a combinatorial problem into an instance of the WDRO problem.

2.1 Optimizing under uncertainty

Stochastic optimization and robust optimization are two classic frameworks to model uncertainty in optimization problems. Distributionally robust optimization is an alternative framework that unifies both approaches.

In this part, we consider an optimization where $\boldsymbol{x} \in \mathcal{X} \subset \{0;1\}^n$ is the decision vector and h is the cost function we want to optimize. This function is subject to uncertainty. We write $\boldsymbol{\xi} \in \Xi \subset \mathbb{R}^n$ the uncertain parameter and $h(\boldsymbol{x}, \boldsymbol{\xi})$ as the objective value of $\boldsymbol{x}$ given a fixed value $\boldsymbol{\xi}$ of uncertainty.

Stochastic optimization In stochastic optimization, we assume that the exact probability distribution P^* of $\boldsymbol{\xi}$ is known. The random variable associated to uncertainties is called $\widetilde{\boldsymbol{\xi}}$. We want to compute a feasible solution $\boldsymbol{x}$ minimizing the expected value of $h(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})$. In other words:

$$\inf_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}_{P^*}[h(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})]$$

Robust optimization In robust optimization, we consider that only the support of the uncertain parameters is known which means that we know all the different values that can be taken by the uncertain parameters. The objective is to compute a feasible solution $\boldsymbol{x}$ minimizing the maximum possible value of $h(\boldsymbol{x}, \boldsymbol{\xi})$.

$$\inf_{\boldsymbol{x} \in \mathcal{X}} \sup_{\boldsymbol{\xi} \in \Xi} h(\boldsymbol{x}, \boldsymbol{\xi})$$

Distributionally robust optimization Distributionally robust optimization is an approach to optimization under uncertainty that assumes only partial distributional information. Unlike the classic approach of stochastic optimization, in DRO, the exact probability distribution P^* is unknown. Instead, we assume that it belongs to an ambiguity set $\mathcal{P}$ of distributions constructed from the partial information. We compute a feasible solution $\boldsymbol{x}$ minimizing the maximum expected value of $h(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})$ among all the possible distributions. In some way, DRO is a robust optimization model where Ξ is replaced by the set $\mathcal{P}$ of distributions.

$$\inf_{\boldsymbol{x} \in \mathcal{X}} \sup_{Q \in \mathcal{P}} \mathbb{E}_Q[h(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})]$$

2.2 Data-driven Wasserstein DRO

The exact probability distribution P^* can be estimated through a finite sample dataset. A natural method is the sample average approximation (SAA) where P^* is replaced by the empirical distribution obtained by averaging of the sample dataset.

Définition 1. Given a list $\widehat{\Xi}$ of N values from Ξ. We define the empirical distribution $\widehat{P}_N$ with $\frac{1}{N}\sum_{\boldsymbol{\xi}\in\widehat{\Xi}}\delta_{\boldsymbol{\xi}}$ where $\delta_{\boldsymbol{\xi}}$ is the dirac distribution at $\boldsymbol{\xi}\in\Xi$.

From this empirical distribution, we build our set of distributions $\mathcal{P}$ with a ball of distributions centered at $\widehat{P}_N$. The metric used to define that ball is the p-Wasserstein distance. We define with $\mathcal{M}(\Xi)$ the set of all probability distributions on Ξ.

Définition 2 (p-Wasserstein distance)**.** Let $Q_1, Q_2 \in \mathcal{M}(\Xi)$ and $1 \leq p \leq +\infty$.

$$d_{W_p}(Q_1, Q_2) = \inf_{\Pi\in\Gamma(Q_1,Q_2)} \int_{\Xi^2} \|\boldsymbol{\xi_1} - \boldsymbol{\xi_2}\|_p \Pi(d\xi_1, d\xi_2)$$

where $\Gamma(Q_1, Q_2)$ is the set of distributions on $\Xi \times \Xi$ with marginal Q_1 and Q_2.

The Wasserstein distance can be used to compare any two distributions Q_1 and Q_2 that can be discrete or continuous. It can be interpreted, as in Figure 1 as the minimum transportation cost for moving from the probability density function of Q_1 to the one of Q_2. The p-norm is used to evaluate the cost of moving some probability from the vector $\boldsymbol{\xi_1}$ to $\boldsymbol{\xi_2}$.

Définition 3 (Wasserstein ball of radius $\varepsilon > 0$ centered at $\widehat{P}_N$)**.**

$$B_{p,\varepsilon}(\widehat{P}_N) := \left\{ Q \in \mathcal{M}(\Xi) : d_{W_p}(\widehat{P}_N, Q) \leq \varepsilon \right\}$$

The Wasserstein ball of radius ε centered at $\widehat{P}_N$ contains all the probability distributions that are at most at a distance of ε using the Wasserstein metric. When $\varepsilon = 0$, it only contains the empirical distribution $\widehat{P}_N$.

2.3 Transform a problem into a WDRO problem

We consider a combinatorial optimization problem Pb written as a 0-1 ILP:

$$\text{Pb} \begin{cases} \inf\limits_{\boldsymbol{x}} \boldsymbol{cx} \\ \text{s.c } \boldsymbol{Ax} \leq \boldsymbol{b} \\ \quad \boldsymbol{x} \in \{0;1\}^n \end{cases}$$

The vector c is subject to uncertainty. As done previously, Ξ is the set of values that can be taken by this parameter and $\widetilde{\xi}$ is the random variable associated to this uncertainty. The exact distribution P^* is not known but a finite sample dataset $\widehat{\Xi}$ is given from which we deduce the empirical distribution $\widehat{P}_N$.

Finally, given a integer $1 \leq p \leq +\infty$ and a real $\varepsilon > 0$, we define the (p, ε)-WDRO-Pb problem.

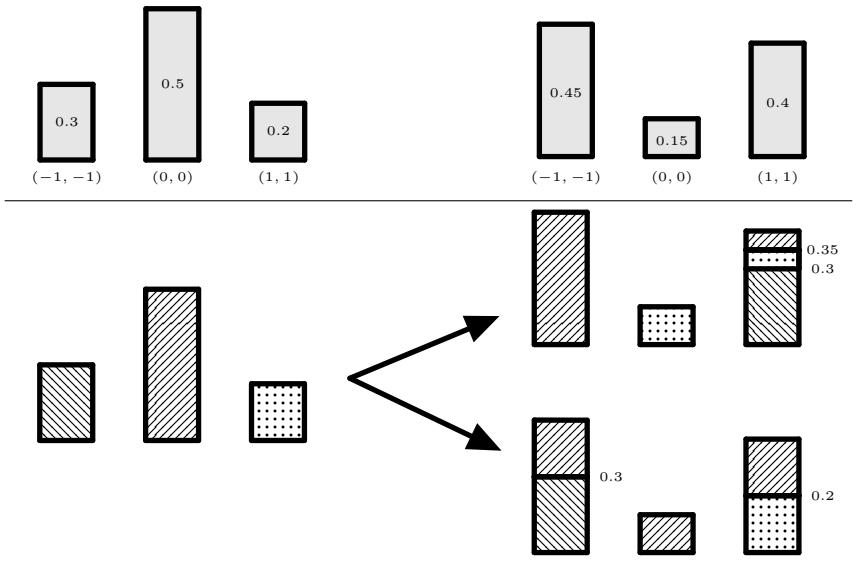

Figure 1: Transformations from histogram Q_1 (upper left) to histogram Q_2 (upper right) representing probabilities of apparition of three vectors $(-1,-1), (0,0)$ and $(1,1)$. Wasserstein distance can be visualized as the best transportation for moving from Q_1 to Q_2. In the three lower histograms, we can see two possible transformations of the histogram Q_1 to histogram Q_2. The second plot seems more efficient since it only moves a portion of the second bar of Q_1, whereas, in the first plot, all the three bars are moved, including the whole second bar. The cost from moving a portion from one bar to another depends on the norm we use. For instance, assuming we use the 2-norm, moving a fraction ε from $(1,1)$ to $(0,0)$ costs $\varepsilon \cdot \|(1,1) - (0,0)\|_2 = \varepsilon\sqrt{2}$. In this example, the cost of the upper transformation would be $0.3\sqrt{8} + 0.15\sqrt{2} + 0.5\sqrt{2} \simeq 1.77$. The cost of the lower transformation would be $0.35\sqrt{2} \simeq 0.5$. The p-Wasserstein distance is the minimum cost (using the p-norm) obtained by the best transformation among all the transformations possible from Q_1 to Q_2. In this case, the distance is at most 0.5.

$$(p,\varepsilon)\text{-WDRO-Pb} \begin{cases} \inf\limits_{\boldsymbol{x}} \sup\limits_{Q} \mathbb{E}_Q[\widetilde{\boldsymbol{\xi}}\boldsymbol{x}] \\ \text{s.c } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ \quad \boldsymbol{x} \in \{0;1\}^n \\ \quad Q \in B_{p,\varepsilon}(\widehat{P}_N) \end{cases}$$

3 Reformulation

We will reformulate the problem by simplifying the expression $\sup_{Q\in B_{p,\varepsilon}(\widehat{P}_N)} \mathbb{E}_Q[\widetilde{\boldsymbol{\xi}}\boldsymbol{x}]$. We first give some intermediate lemmas.

Lemme 3.1. *Let P and Q be two distributions on Ξ. Then $\|\mathbb{E}_P[\widetilde{\boldsymbol{\xi}}] - \mathbb{E}_Q[\widetilde{\boldsymbol{\xi}}]\|_p \leq d_{W_p}(P,Q)$*

Proof. We recall that $d_{W_p}(Q_1,Q_2) = \inf_{\Pi\in\Gamma(P,Q)} \int_{\Xi^2} \|\boldsymbol{\xi_1} - \boldsymbol{\xi_2}\|_p \Pi(d\xi_1, d\xi_2)$

Let $\Pi \in \Gamma(P,Q)$
$\mathbb{E}_P[\widetilde{\boldsymbol{\xi}}] = \int_\Xi \boldsymbol{\xi}\ P(d\boldsymbol{\xi}) = \int_{\Xi^2} \boldsymbol{\xi}_1\ \Pi(d\boldsymbol{\xi}_1, d\boldsymbol{\xi}_2)$
$\mathbb{E}_Q[\widetilde{\boldsymbol{\xi}}] = \int_\Xi \boldsymbol{\xi}\ Q(d\boldsymbol{\xi}) = \int_{\Xi^2} \boldsymbol{\xi}_2\ \Pi(d\boldsymbol{\xi}_1, d\boldsymbol{\xi}_2)$

$$\begin{aligned}\|\mathbb{E}_P[\widetilde{\boldsymbol{\xi}}] - \mathbb{E}_Q[\widetilde{\boldsymbol{\xi}}]\|_p &= \Big\| \int_{\Xi^2} (\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2)\ \Pi(d\boldsymbol{\xi}_1, d\boldsymbol{\xi}_2)\Big\|_p \\ &\leq \int_{\Xi^2} \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|_p\ \Pi(d\boldsymbol{\xi}_1, d\boldsymbol{\xi}_2) \\ &\leq d_{W_p}(P,Q)\end{aligned}$$ □

Lemme 3.2. $\sup_{Q\in B_{p,\varepsilon}(\widehat{P}_N)} \mathbb{E}_Q[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] = \sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} \mathbb{E}_{P^*}[\boldsymbol{x}(\widetilde{\boldsymbol{\xi}} + \boldsymbol{\Delta})]$

Proof. We first prove that $\sup_{Q\in B_{p,\varepsilon}(\widehat{P}_N)} \mathbb{E}_Q[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] \leq \sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} \mathbb{E}_{P^*}[\boldsymbol{x}(\widetilde{\boldsymbol{\xi}} + \boldsymbol{\Delta})]$ and then the second inequality.

Let $Q \in B_{p,\varepsilon}(\widehat{P}_N)$.

$$\begin{aligned}\mathbb{E}_Q[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] &= \mathbb{E}_{P^*}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] + \mathbb{E}_Q[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] - \mathbb{E}_{P^*}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] \\ &= \mathbb{E}_{P^*}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] + \boldsymbol{x}(\mathbb{E}_Q[\widetilde{\boldsymbol{\xi}}] - \mathbb{E}_{P^*}[\widetilde{\boldsymbol{\xi}}])\end{aligned}$$

We have $\|\mathbb{E}_Q[\widetilde{\boldsymbol{\xi}}] - \mathbb{E}_{P^*}[\widetilde{\boldsymbol{\xi}}]\|_p \leq \varepsilon$ using Lemma 3.1. So

$$\begin{aligned}&\leq \sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} \mathbb{E}_{P^*}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] + \boldsymbol{x}\Delta \\ &\leq \sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} \mathbb{E}_{P^*}[\boldsymbol{x}(\widetilde{\boldsymbol{\xi}} + \boldsymbol{\Delta})]\end{aligned}$$

As this inequality is true for any $Q \in B_{p,\varepsilon}(\widehat{P}_N)$, it results

$$\sup_{Q\in B_{p,\varepsilon}(\widehat{P}_N)} \mathbb{E}_Q[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] \leq \sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} \mathbb{E}_{P^*}[\boldsymbol{x}(\widetilde{\boldsymbol{\xi}} + \boldsymbol{\Delta})]$$

The first inequality is then proved. Let now $\|\boldsymbol{\Delta}\|_p \leq \varepsilon$, with $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_n)$. We show that we can construct a distribution Q such that its Wasserstein distance to P^* is less than ε.

Let $Q = \frac{1}{N}\sum_{\boldsymbol{\xi}\in\widehat{\Xi}} \delta_{\boldsymbol{\xi}+\boldsymbol{\Delta}}$. The Wasserstein distance between P^* and Q is the optimal transport needed to obtain Q from P^*. One way to obtain Q from P^* is to move $\boldsymbol{\xi}$ to $\boldsymbol{\xi} + \boldsymbol{\Delta}$ for each component $\boldsymbol{\xi} \in \widehat{\Xi}$. So we obtain :

$$d_{W_p}(P^*, Q) \leq \frac{1}{N}\sum_{\boldsymbol{\xi}\in\widehat{\Xi}} d(\delta_{\boldsymbol{\xi}}, \delta_{\boldsymbol{\xi}+\boldsymbol{\Delta}}) \leq \frac{1}{N}\sum_{\boldsymbol{\xi}\in\widehat{\Xi}} \|\boldsymbol{\Delta}\|_p \leq \|\boldsymbol{\Delta}\|_p \leq \varepsilon$$

Thus $Q \in B_{p,\varepsilon}(\widehat{P}_N)$. In addition,

$$\mathbb{E}_{P^*}[\boldsymbol{x}(\widetilde{\boldsymbol{\xi}} + \boldsymbol{\Delta})] = \mathbb{E}_{P^*}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] + \boldsymbol{x}\boldsymbol{\Delta} = \frac{1}{N}\sum_{\boldsymbol{\xi}\in\widehat{\Xi}} \boldsymbol{x}\boldsymbol{\xi} + \boldsymbol{x}\boldsymbol{\Delta}$$

$$\mathbb{E}_{Q}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] = \frac{1}{N}\sum_{\boldsymbol{\xi}\in\widehat{\Xi}} \boldsymbol{x}(\boldsymbol{\xi} + \boldsymbol{\Delta}) = \frac{1}{N}\sum_{\boldsymbol{\xi}\in\widehat{\Xi}} \boldsymbol{x}\boldsymbol{\xi} + \boldsymbol{x}\boldsymbol{\Delta}$$

As $Q \in B_{p,\varepsilon}(\widehat{P}_N)$,

$$\mathbb{E}_{P^*}[\boldsymbol{x}(\widetilde{\boldsymbol{\xi}} + \boldsymbol{\Delta})] \leq \sup_{Q\in B_{p,\varepsilon}(\widehat{P}_N)} \mathbb{E}_{Q}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}]$$

As this is true for every vector $\boldsymbol{\Delta}$ with $\|\boldsymbol{\Delta}\|_p \leq \varepsilon$,

$$\sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} \mathbb{E}_{P^*}[\boldsymbol{x}(\widetilde{\boldsymbol{\xi}} + \boldsymbol{\Delta})] \leq \sup_{Q\in B_{p,\varepsilon}(\widehat{P}_N)} \mathbb{E}_{Q}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] \qquad \square$$

We now prove our main theorem that reformulates the probabilistic objective function of (p,ε)-WDRO-Pb into a deterministic objective function. Recall that the dual norm of $\|.\|_p$ is $\|.\|_q$ with q such as $\frac{1}{p}+\frac{1}{q}=1$ (p (resp q) can be infinite if $q=1$ (resp. $p=1$)) which means that $\|\boldsymbol{z}\|_q = \sup_{\boldsymbol{x}}\{\boldsymbol{z}\boldsymbol{x} \mid \|\boldsymbol{x}\|_p \leq 1\}$.

Théorème 3.1. $\inf\limits_{\substack{\boldsymbol{x}\in[\![0;1]\!]^n \\ \boldsymbol{A}\boldsymbol{x}\leq\boldsymbol{b}}} \sup\limits_{Q\in B_{p,\varepsilon}(\widehat{P}_N)} \mathbb{E}_{Q}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] = \inf\limits_{\substack{\boldsymbol{x}\in[\![0;1]\!]^n \\ \boldsymbol{A}\boldsymbol{x}\leq\boldsymbol{b}}} \|\boldsymbol{x}\|_q \cdot \varepsilon + \frac{1}{N}\sum\limits_{\boldsymbol{\xi}\in\widehat{\Xi}} \boldsymbol{x}\boldsymbol{\xi}$

Proof. Let $\boldsymbol{x} \in [\![0;1]\!]^n$ such that $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$. By Lemma 3.2,

$$\begin{aligned}
\sup_{Q\in B_{p,\varepsilon}(\widehat{P}_N)} \mathbb{E}_{Q}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] &= \sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} \mathbb{E}_{P^*}[\boldsymbol{x}(\widetilde{\boldsymbol{\xi}} + \boldsymbol{\Delta})] \\
&= \sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} (\mathbb{E}_{P^*}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] + \mathbb{E}_{P^*}[\boldsymbol{x}\boldsymbol{\Delta}]) \\
&= \mathbb{E}_{P^*}[\boldsymbol{x}\widetilde{\boldsymbol{\xi}}] + \sup_{\|\boldsymbol{\Delta}\|_p\leq\varepsilon} \boldsymbol{x}\boldsymbol{\Delta} \\
&= \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{x}\boldsymbol{\xi} + \sup_{\|\boldsymbol{\Theta}\|_p\leq 1} \boldsymbol{x}\boldsymbol{\Theta}\cdot\varepsilon
\end{aligned}$$

By the definition of dual norm

$$= \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{x}\boldsymbol{\xi} + \|\boldsymbol{x}\|_q \cdot \varepsilon \qquad \square$$

Remarque 1. Note that this result can also be shown using the duality theory from the original problem. It is a special case of the reformulation done in [5].

Remarque 2. The new term $\|\boldsymbol{x}\|_q$ added from the reformulation can be seen as a penalty from the number of uncertain elements we choose in our solution. The more uncertain elements we use to construct our solution, the more penalty we get from this term $\|\boldsymbol{x}\|_q$. What this reformulation means is that we want a good trade-off between the evaluation of a solution using the past data and its uncertainty.

4 Main algorithm

In the following section, we note $\mathcal{J} = (\boldsymbol{A}, \boldsymbol{b}, P)$ an instance of (p, ε)-WDRO-Pb where $\boldsymbol{A}, \boldsymbol{b}$ are the constraint coefficients matrix and vector which defines the set of feasible solutions and P is the empirical distribution, which is the center of the distribution ambiguity set. We also note $\mathcal{I} = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$ an instance of Pb where $\boldsymbol{A}, \boldsymbol{b}$ are the constraint coefficients matrix and vector which defines the set of feasible solutions and $\boldsymbol{c}$ is the cost coeffients vector.

Theorem 3.1 shows that (p, ε)-WDRO-Pb can be reformulated into a problem with a deterministic objective function, without any notion of probability distributions.

$$(p,\varepsilon)\text{-WDRO-Pb} \begin{cases} \inf\limits_{\boldsymbol{x}} \|\boldsymbol{x}\|_q \cdot \varepsilon + \dfrac{1}{N} \displaystyle\sum_{\xi \in \widehat{\Xi}} \boldsymbol{x}\boldsymbol{\xi} \\ \text{s.c } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ \quad \boldsymbol{x} \in \{0;1\}^n \end{cases}$$

As $\boldsymbol{x}$ is a binary vector, we can rewrite $\|\boldsymbol{x}\|_q = \sqrt[q]{\sum_i x_i^q} = \sqrt[q]{\sum_i x_i} = \sqrt[q]{|\boldsymbol{x}|}$ where $|\boldsymbol{x}| = \sum_i x_i$.

Suppose that an algorithm $\mathcal{A}$ can return a solution of (Pb). We provide a method to solve its WDRO counterpart using the same algorithm $\mathcal{A}$. Remark that unlike (Pb), the objective function in (p, ε)-WDRO-Pb is not linear. Solving such a problem can be hard in general cases. The idea of our method is to linearize the non linear part of the objective function. To do so, we want to replace $f(|\boldsymbol{x}|) = \sqrt[q]{|\boldsymbol{x}|}$ by tangents $g_i : k \mapsto a_i k + b_i$ where g_i is the tangent of f at value $|\boldsymbol{x}| = i$. Particular attention was paid on the polynomiality of our method and computing those coefficients can be potentially hard. So we first prove that it is possible to compute coefficients that are close enough to the real coefficients. Those coefficients have the same properties we need than the real tangent coefficients. When the objective function is linearized, we are able to solve it using the algorithm $\mathcal{A}$. This procedure is described with Algorithm 1. We write $\lfloor r \rfloor_k$ the real r rounded to k decimals.

Lemme 4.1. *Assuming $\mathcal{A}$ is a polynomial algorithm, the complexity of Algorithm 1 is also polynomial.*

Proof. If $p \neq 1$, Algorithm 1 makes $n+1$ iterations of a for loop. Each iteration performs the following operations.

- It builds a_i by computing the θ-th decimals of $\sqrt[q]{i+1}$ and $\sqrt[q]{i}$. Each decimal of $\sqrt[q]{x}$ can be returned by comparing $(\lfloor \sqrt[q]{x} \rfloor_{\theta'-1} + d \cdot 10^{-\theta'})^q$ and x for all $d \leq 9$ and $\theta' \leq \theta$. Thus a_i is computed in polynomial time.
- It runs one time the algorithm $\mathcal{A}$, in polynomial time with polynomial size inputs.
- It compares $\omega_i + \varepsilon \cdot \sqrt[q]{i}$ and ω which is of the form $\omega_j + \varepsilon \cdot \sqrt[q]{j}$ for some $j < i$ where ω_i and ω_j are (polynomial size) rationals. This comparison can be done in polynomial time [2].

On the other hand, if $p = 1$ then the algorithm runs one time $\mathcal{A}$ and does one comparison in polynomial time. □

Algorithm 1 Algorithm to solve (p,ε)-WDRO-Pb

Require: An algorithm $\mathcal{A}$ for Pb returning a feasible solution and an instance $\mathcal{J} = (\boldsymbol{A}, \boldsymbol{b}, \widehat{P}_N)$ of (p,ε)-WDRO-Pb
Ensure: A feasible solution $\boldsymbol{x}$ of $\mathcal{J}$

```
if p = 1 then
    x ← A(A, b, c = (1/N) Σ_{ξ∈Ξ̂} ξ)
    return x if c·x + ε < 0 and the vector 0 otherwise
else
    q ← p/(p−1)
    ω ← +∞
    x ← 0
    θ ← ⌈log10(4·q²(n+1)⁶)⌉
    for i from 0 to n do
        a_i ← ⌊∛(i+1)⌋_θ − ⌊∛i⌋_θ + 10^{−θ} .
        x^(i) ← A(A, b, c = (1/N) Σ_{ξ∈Ξ̂} ξ + ε a_i · 1)
        ω_i ← (1/N) Σ_{ξ∈Ξ̂} ξ · x^(i)
        if ω_i + ε · ∛i < ω then
            ω ← ω_i + ε · ∛i
            x ← x^(i)
    return x
```

In order to prove the correctness, we first prove the following useful lemmas.

Lemme 4.2. *For every $i < j \in [\![0;n]\!]$, $q \geq 1$ and $r \geq 1$, $\sqrt[q]{j}^r - \sqrt[q]{i}^r \geq \frac{j-i}{qn^2}$.*

Proof. If, firstly $i = 0$ then $j \geq 1$ and $\sqrt[q]{j}^r - \sqrt[q]{i}^r \geq 1 \geq \frac{j}{qn^2}$. On the other hand, if $i \geq 1$, recall that, if $x > y \geq 1$ then $x^r - y^r \geq x - y$. In that case, we then prove the lemma only for $r = 1$. We use the identity $x^q - y^q = (x-y) \cdot \sum_{k=0}^{q-1} x^k \cdot y^{q-k-1}$ which leads to

$$\sqrt[q]{j} - \sqrt[q]{i} = \frac{j-i}{\sum_{k=0}^{q-1} \sqrt[q]{j}^k \cdot \sqrt[q]{i}^{q-k-1}} \geq \frac{j-i}{\sum_{k=0}^{q-1} j \cdot i} \geq \frac{j-i}{qn^2} \quad \square$$

Lemme 4.3. *For every $i, j \in [\![0;n]\!]$, $\sqrt[q]{i} + a_i \cdot (j-i) \geq \sqrt[q]{j}$.*

Proof. As $i \mapsto \sqrt[q]{i}$ is concave, proving that the lemma is true for $j \in \{i-1, i+1\}$ is sufficient.

For $j = i+1$, we have

$$\sqrt[q]{i} + a_i \cdot (j-i) = \sqrt[q]{i} + \left\lfloor \sqrt[q]{i+1} \right\rfloor_\theta - \left\lfloor \sqrt[q]{i} \right\rfloor_\theta + 10^{-\theta}$$

As $\lfloor x \rfloor_\theta \in [x - 10^{-\theta}; x]$

$$\geq \sqrt[q]{i} + \sqrt[q]{i+1} - 10^{\theta} - \sqrt[q]{i} + 10^{-\theta} \geq \sqrt[q]{i+1}$$

For $j = i-1$, assuming $i > 0$, we first prove the following intermediate result: $(\sqrt[q]{i} - \sqrt[q]{i-1}) - (\sqrt[q]{i+1} - \sqrt[q]{i}) \geq \frac{1}{q^2(n+1)^6}$. We use again the formula $x^q - y^q = (x-y) \cdot \sum_{k=0}^{q-1} x^k y^{q-k-1}$. Thus,

$$\begin{aligned}
(\sqrt[q]{i} - \sqrt[q]{i-1}) - (\sqrt[q]{i+1} - \sqrt[q]{i}) &= \frac{1}{\sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i-1}^{q-k-1}} - \frac{1}{\sum_{k=0}^{q-1} \sqrt[q]{i+1}^k \sqrt[q]{i}^{q-k-1}} \\
&= \frac{1}{\sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i-1}^{q-k-1}} - \frac{1}{\sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i+1}^{q-k-1}} \\
&= \frac{\sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i+1}^{q-k-1} - \sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i-1}^{q-k-1}}{\sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i-1}^{q-k-1} \cdot \sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i+1}^{q-k-1}} \\
&= \frac{\sum_{k=0}^{q-2} \sqrt[q]{i}^k \cdot (\sqrt[q]{i+1}^{q-k-1} - \sqrt[q]{i-1}^{q-k-1})}{\sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i-1}^{q-k-1} \cdot \sum_{k=0}^{q-1} \sqrt[q]{i}^k \sqrt[q]{i+1}^{q-k-1}}
\end{aligned}$$

As $i > 0$, $i \leq n$ and by Lemma 4.2

$$\begin{aligned}
&\geq \frac{\sum_{k=0}^{q-2} 2/(q(n+1)^2)}{(qn^2) \cdot (q(n+1)^2)} = \frac{2 \cdot (q-1)/(q(n+1)^2)}{(qn^2) \cdot (q(n+1)^2)} \\
&\geq \frac{1}{q^2(n+1)^6}
\end{aligned}$$

Recall that, in Algorithm 1, θ is set to $\lceil \log_{10}(4 \cdot q^2(n+1)^6) \rceil \geq \log_{10}(4 \cdot q^2(n+1)^6)$. Then $4 \cdot 10^{-\theta} \leq \frac{1}{q^2(n+1)^6}$. Consequently

$$\begin{aligned}
\sqrt[q]{i} - \sqrt[q]{i-1} &\geq \sqrt[q]{i+1} - \sqrt[q]{i} + 4 \cdot 10^{-\theta} \\
\left\lfloor \sqrt[q]{i} \right\rfloor_\theta - \left\lfloor \sqrt[q]{i-1} \right\rfloor_\theta + 10^{-\theta} &\geq \left\lfloor \sqrt[q]{i+1} \right\rfloor_\theta - \left\lfloor \sqrt[q]{i} \right\rfloor_\theta - 10^{-\theta} + 4 \cdot 10^{-\theta} \\
2 \cdot \left\lfloor \sqrt[q]{i} \right\rfloor_\theta - \left\lfloor \sqrt[q]{i+1} \right\rfloor_\theta - 10^{-\theta} &\geq \left\lfloor \sqrt[q]{i-1} \right\rfloor_\theta + \cdot 10^{-\theta} \\
\left\lfloor \sqrt[q]{i} \right\rfloor_\theta - a_i &\geq \left\lfloor \sqrt[q]{i-1} \right\rfloor_\theta + \cdot 10^{-\theta} \\
\sqrt[q]{i} - a_i &\geq \sqrt[q]{i-1}
\end{aligned}$$

□

Lemme 4.4. *Let $b_i = \sqrt[q]{i} - a_i \cdot i$. Let $\mathcal{J} = (\boldsymbol{A}, \boldsymbol{b}, \widehat{P}_N)$ be an instance of (p, ε)-WDRO-Pb. We assume $p \neq 1$, let $\boldsymbol{x}^*$ be an optimal solution of $\mathcal{J}$ with value ω^* and, for $i \in [\![0; n]\!]$, let $\boldsymbol{x}_i^*$ be an optimal solution of $\mathcal{I}_i$, the instance of (Pb) defined as $\mathcal{I}_i = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c} = \frac{1}{N} \sum_{\xi \in \widehat{\Xi}} \boldsymbol{\xi} + \varepsilon \cdot a_i \cdot \mathbf{1})$ and let ω_i^* be its optimal value. Then $\min_{i \in [\![0;n]\!]} \omega_i^* + \varepsilon \cdot b_i = \omega^*$.*

Proof. First, $\omega^* = \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x^*} + \varepsilon\cdot\sqrt[q]{|\boldsymbol{x^*}|} = \dfrac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x^*} + a_{|x^*|}\cdot|x^*| + \varepsilon\cdot b_{|x^*|}$. Note that $\frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x^*} + a_{|x^*|}\cdot|x^*|$ is the objective value of x^* in $\mathcal{I}_{|x^*|}$ then is greater than $\omega_{|x^*|}$. Then $\omega^* \geq \omega_{|x^*|} + \varepsilon b_{|x^*|} \geq \min_{i\in[\![0;n]\!]} \omega_i^* + \varepsilon\cdot b_i$.

We now prove the second inequality. Let $g_i : j \mapsto a_i\cdot j + b_i$. Lemma 4.3 shows that for $i, j \in [\![0;n]\!]$, $g_i(j) = a_i\cdot j + \sqrt[q]{i} - a_i\cdot i \geq \sqrt[q]{j}$. Then, given $x\in\{0;1\}^n$, we know that $\sqrt[q]{|\boldsymbol{x}|} \leq a_i|\boldsymbol{x}| + b_i$ for any $i\in[\![0;n]\!]$.

Suppose that $\boldsymbol{x}_i^*$ is such that $|\boldsymbol{x}_i^*| = j \neq i$. Then solving $\mathcal{I}_j$ gives a better solution for $\mathcal{J}$ than $\boldsymbol{x}_i^*$, in other words: $\omega_j^* + \varepsilon\cdot b_j \leq \omega_i^* + \varepsilon\cdot b_i$.

$$\omega_j^* + \varepsilon\cdot b_j = \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x_j^*} + \varepsilon\cdot a_j\cdot|\boldsymbol{x}_j^*| + \varepsilon\cdot(\sqrt[q]{j} - a_j\cdot j)$$

By optimality of $\boldsymbol{x_j^*}$

$$\begin{aligned}
&\leq \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x_i^*} + \varepsilon\cdot a_j\cdot|\boldsymbol{x}_i^*| + \varepsilon\cdot(\sqrt[q]{j} - a_j\cdot j)\\
&\leq \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x_i^*} + \varepsilon\cdot a_j\cdot|\boldsymbol{x}_i^*| + \varepsilon\cdot(a_i\cdot j + \sqrt[q]{i} - a_i\cdot i - a_j\cdot j)\\
&\leq \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x_i^*} + \varepsilon\cdot a_i\cdot|\boldsymbol{x}_i^*| + \varepsilon\cdot b_i = \omega_i^* + \varepsilon\cdot b_i
\end{aligned}$$

Thus, if $j = \arg\min_{i\in[\![0;n]\!]} \omega_i^* + \varepsilon\cdot b_i$, then $|\boldsymbol{x_j}^*| = j$. We then deduce that $\omega_j^* + \varepsilon\cdot b_j = \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x_j^*} + \varepsilon\cdot a_j\cdot|\boldsymbol{x}_j^*| + \varepsilon\cdot(\sqrt[q]{j} - a_j\cdot j) = \frac{1}{N}\sum_{\xi\in\widehat{\Xi}} \boldsymbol{\xi}\cdot\boldsymbol{x_j^*} + \varepsilon\cdot\sqrt[q]{j}$ which is the objective value of $\boldsymbol{x_j^*}$ in $\mathcal{J}$, thus is greater than ω^*. □

Théorème 4.1. *If Pb is α-approximable in polynomial time, then, for any $p \in \mathbb{R}_+^* \cup \{+\infty\}$ and $\varepsilon > 0$, (p, ε)-WDRO-Pb is α-approximable in polynomial time.*

Proof. Let $\mathcal{A}$ be a polynomial α-approximation algorithm for (Pb) and $\mathcal{J} = (\boldsymbol{A}, \boldsymbol{b}, \widehat{P}_N)$ an instance of (p, ε)-WDRO-Pb. We keep the same notation as Lemma 4.4.

If $p = 1$, we note $\boldsymbol{x}$ the solution given by $\mathcal{A}(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c} = \frac{1}{N} \sum_{\xi \in \widehat{\Xi}} \boldsymbol{\xi})$ and x^* an optimal solution of the associated instance. We have $\boldsymbol{c} \cdot \boldsymbol{x^*} \leq \boldsymbol{c} \cdot \boldsymbol{x} \leq \alpha \boldsymbol{c} \cdot \boldsymbol{x^*}$ because $\mathcal{A}$ is α-approximable. If $\boldsymbol{c} \cdot \boldsymbol{x} + \varepsilon < 0$, we obtain $\boldsymbol{c} \cdot \boldsymbol{x^*} + \varepsilon \leq \boldsymbol{c} \cdot \boldsymbol{x} + \varepsilon \leq \alpha \boldsymbol{c} \cdot \boldsymbol{x^*} + \varepsilon \leq \alpha(\boldsymbol{c} \cdot \boldsymbol{x^*} + \varepsilon)$. If $\boldsymbol{c} \cdot \boldsymbol{x} + \varepsilon \geq 0$, the optimal solution is $\mathbf{0}$, the zero vector, which is also what Algorithm 1 returns. Thus, Algorithm 1 is α-approximable for (p, ε)-WDRO-Pb if $p = 1$.

If $p \neq 1$, at each iteration i of Algorithm 1, $\boldsymbol{x^{(i)}}$ is a solution of $\mathcal{I}_i$ with value ω_i given by the algorithm $\mathcal{A}$. Since $\mathcal{A}$ is an α-approximation, $\omega_i^* \leq \omega_i \leq \alpha \omega_i^*$.

We showed that $\min_{i \in [\![0;n]\!]} \omega_i^* = \omega^*$ in Lemma 4.4. Since $\omega_i^* \leq \omega_i \leq \alpha \omega_i^*$ holds for any $i \in [\![0; n]\!]$, it proves that $\omega^* \leq \min_{i \in [\![0;n]\!]} \omega_i \leq \alpha \omega^*$ thus, Algorithm 1 is an α-approximation for (p, ε)-WDRO-Pb. By Lemma 4.1, this approximation is polynomial. □

Remarque 3. Note that when the decision vector of (Pb) is a vector of bounded integers instead of binaries, we can apply the same idea of Algorithm 1 to obtain a pseudo-polynomial time complexity algorithm to solve (p, ε)-WDRO-Pb. Indeed, the only parameter that changes is the number of states of $|\boldsymbol{x}|$ for $\boldsymbol{x} \in [\![0; M]\!]^n$. When the decision vector is binary, $0 \leq |\boldsymbol{x}| \leq n$ and when it is a bounded integer between 0 and M, $0 \leq |\boldsymbol{x}| \leq nM$.

Remarque 4. When our framework is applied to the shortest path problem, we obtain a polynomial distributionally robust shortest path problem whereas most of the robust versions of the shortest path problem are known to be NP-Hard. This result is similar to the one in [1]. This is due to the fact that, using Wasserstein metric, the worst case distribution is usually easy to compute.

5 Conclusion and perspectives

In this paper, we describe distributionally robust optimization paradigm to take account of uncertainties, especially the data-driven Wasserstein distributionally robust optimization framework. Knowing a black box algorithm to solve a 0-1 discrete optimization problem, we propose an algorithm to solve its distributionally robust counterpart. Complexity results related to polynomiality of the original problem are still available for the new problem. We are aware that the purpose of our algorithm is only to give a strong theoretical complexity result on the WDRO problems. From a practical point of view, this algorithm should obviously be adapted to each combinatorial optimization problem on a case-by-case basis. For example, adapting the Dijkstra algorithm should provide a better time complexity than our Algorithm 1 for the WDRO shortest path problem.

An interesting perspective is to assume that the constraint coefficients are subject to uncertainty. In this case, we have to describe another framework that

can take account of the modification of the feasible solution set by the uncertain coefficients. For example, one paradigm that can be used is the distributionally robust chance constrained programs studied in [3, 7, 10].

References

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

[2] Burnikel, C., Funke, S., Mehlhorn, K., Schirra, S., and Schmitt, S. A separation bound for real algebraic expressions. In *European Symposium on Algorithms* (2001), Springer, pp. 254–265.

[3] Chen, Z., Kuhn, D., and Wiesemann, W. Data-driven chance constrained programs over wasserstein balls, 2018.

[4] Delage, E., and Ye, Y. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research 58* (06 2010), 595–612.

[5] Esfahani, P. M., and Kuhn, D. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations, 2017.

[6] Gao, R., and Kleywegt, A. J. Distributionally robust stochastic optimization with wasserstein distance, 2016.

[7] Ji, R., and Lejeune, M. A. Data-driven distributionally robust chance-constrained optimization with wasserstein metric. *Journal of Global Optimization 79*, 4 (2021), 779–811.

[8] Kuhn, D., Esfahani, P. M., Nguyen, V. A., and Shafieezadeh-Abadeh, S. Wasserstein distributionally robust optimization: Theory and applications in machine learning, 2019.

[9] Rahimian, H., and Mehrotra, S. Distributionally robust optimization: A review, 2019.

[10] Xie, W. On distributionally robust chance constrained programs with wasserstein distance. *Mathematical Programming 186*, 1 (2021), 115–155.