

Received May 29, 2021, accepted June 11, 2021, date of publication June 22, 2021, date of current version July 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3091432

# Consensus Distributionally Robust Optimization With Phi-Divergence

SHUNICHI OHMORI 

Department of Business Design and Management, Waseda University, Tokyo 169-8555, Japan

e-mail: ohmori0406@aoni.waseda.jp

This work was supported by the Japan Society for the promotion of Science (JSPS) KAKENHI under Grant 19K04894.

**ABSTRACT** We study an efficient algorithm to solve the distributionally robust optimization (DRO) problem, which has recently attracted attention as a new paradigm for decision making in uncertain situations. In traditional stochastic programming, a decision is sought that minimizes the expected cost over the probability distribution of the unknown parameters. In contrast, in DRO, robust decision making can be derived from data without assuming a probability distribution; thus, it is expected to provide a powerful method for data-driven decision making. However, it is computationally difficult to solve the DRO problem and even by state-of-art solvers the problem size that can be solved to optimality is still limited. Therefore, we propose an efficient algorithm for solving DRO based on consensus optimization (CO). CO is a distributed algorithm in which a large-scale problem is decomposed into smaller subproblems. Because different local solutions are obtained by solving subproblems, a consensus constraint is imposed to ensure that these solutions are equal, thereby guaranteeing global convergence. We applied the proposed method to linear programming, quadratic programming, and second-order cone programming in numerical experiments and verified its effectiveness.

**INDEX TERMS** Alternating direction method of multipliers, consensus optimization, decomposition method, distributionally robust optimization, stochastic programming.

## I. INTRODUCTION

The effects of uncertainty in decision making continue to increase in the dynamic and volatile business environment. In recent years, the use of big data has created new possibilities for decision making in such uncertain situations.


Stochastic programming has been studied extensively as a traditional decision-making problem in uncertain situations. In stochastic programming, given a probability distribution  $p(u)$  of unknown parameters  $u$ , a decision  $x$  is sought that minimizes the expected cost  $\mathbb{E}f(x, u)$ . In the real world, the probability distribution is unknown and must be inferred from data. However, several difficulties arise, as described below.

First, when optimizing an unknown true objective function that is estimated from observed data that are subject to random error, even if the value estimates are unbiased, the uncertainty in these estimates together with the optimization-based selection process results in the value estimates for the recommended action having high bias. This means that the

resultant out-of-sample performance is often disappointing. In the field of decision analysis, this is known as the *optimizer's curse* [1]. Second, even if the probability distribution  $p(u)$  and decision  $x$  are provided, the calculation of the expected value  $\mathbb{E}(x, u)$  requires multiple integrals, which is # P-hard. Third, assumptions and validations of probability distribution, which might involve the selection of distribution model and parameter estimation, can take long time. Owing to the need for quicker decision making and the shortage of data scientists, autonomous data-driven decision making has attracted significant practical interest.

Bertsimas and Kallus [2] noted that probability distribution is imaginary, based on human assumption, and is never observed in practice. Furthermore, data always really exist and are observable. Therefore, they claimed that data-driven decision making without the explicit consideration of the probability distribution should be appropriate, as it enables decisions to be made based on evidence rather than assumption.

Data-driven stochastic programming is an alternative paradigm in which the probability distribution of uncertain parameters is not known, and instead, the realization of data

The associate editor coordinating the review of this manuscript and approving it for publication was Jamshid Aghaei .

is provided. The data are composed of the training data  $\mathcal{U}_T$  and validation data  $\mathcal{U}_V$ . In the training phase, the data-driven decision  $\hat{x}_T$  and certificate  $\hat{z}_T = \mathbb{E}[f(\hat{x}_T, u|\mathcal{U}_T)]$  are output, whereas in the validation phase, the data-driven decision of the out-of-sample performance  $\hat{z}_V = \mathbb{E}[f(\hat{x}_T, u|\mathcal{U}_V)]$  is evaluated.

Distributionally robust optimization (DRO) has attracted attention as an approach for data-driven stochastic programming in recent years. DRO assumes that the probability distribution lies within the *ambiguity set*  $\mathcal{P}$ , which is inferred from the data. Under this assumption, the decision  $x$  is sought to minimize the worst-case expected cost over an ambiguity set, *i.e.*,  $\sup_{p \in \mathcal{P}} \mathbb{E}_p f(x, u)$ . DRO exhibits several good properties [3]. First, the worst-case approach mitigates the optimizer's curse and often leads to better out-of-sample performance than the *sample average approximation* (SAA), in which the probability distribution is approximated by the empirical discrete distribution. Second, the theoretical guarantee of the out-of-sample performance, which is the probability of no over-fitting  $\mathbf{prob}(\hat{z}_T \leq \hat{z}_V)$  occurring, is derived. Third, asymptotic optimality is proven; that is, as the number of samples approaches to infinity, the data-driven solution converges to the true optimal solution of the problem. Finally, the DRO problem can be transformed into a convex programming problem, and thus, it can be solved in polynomial time.

However, unlike the case in theory, the DRO problem often takes a very long computation time to solve because it is a nonlinear convex programming problem. In particular, in recent years, the problem size has become large in terms of both the sample size and dimensionality. As an illustrative example, Table 1 presents the calculation time of DRO using the ambiguity set derived from Kullback–Leibler (KL) divergence for a linear programming (LP) problem with variables  $n = 200$  and constraints  $m = 300$ . Refer to Section III for details of the model and Section V for the experimental environment. The upper bound of the calculation time is set to 2 hours (h), and if this time is exceeded, it is set as not applicable (N/A). According to Table 1, It is possible to solve the problem in a short time up to a sample size of  $N = 10^4$ . However, when the sample size is  $N = 10^5$ , the computation time increases drastically and the solution cannot be calculated within 2 h. Based on this observation, even if the original problem is simple, the calculation time in DRO will be within an unacceptable range owing to the increase in the sample size. As computational responsiveness is very important for timely decision making in the rapidly changing business environment, it is necessary to develop an algorithm that can solve the problem efficiently with large-scale data.

In this study, we investigate an efficient algorithm to solve the DRO. We propose a distributed optimization technique that uses consensus optimization (CO), which has gained popularity as a decomposition method for large-scale problems in the convex programming. CO is a type of alternating direction method of multipliers (ADMM), which divides a large-scale problem into multiple small-scale subproblems and solves them in a distributed manner. As each subproblem

**TABLE 1.** Computation time of DRO for LP (variables  $n = 200$ , constraints  $m = 300$ ).

sample size $N$	DRO
$10^1$	0.03
$10^2$	0.05
$10^3$	1.69
$10^4$	157
$10^5$	N/A

results in a different local solution, a consensus constraint is imposed that ensures that these solutions are equal. Although this CO method is inferior to second-order algorithms such as the interior-point method in terms of the convergence rate, its convergence is fast in many applications if high accuracy is not required.

More precisely, the optimization methods for convex programming iteratively solve the approximation problems to generate a sequence of points that converge to the optimal solution. Therefore, the calculation time is dependent on (1) the per-iteration solution time of the approximation problem and (2) the number of iterations until convergence. In general, a trade-off exists between the two; thus, it is important to use the appropriate algorithm according to the problem structure.

In the second-order method such as interior-point method, the number of iterations is small because the approximation accuracy is high, but a computation time of at least  $O((n + m + N)^3)$  is required to solve the approximation problem, where  $n$  is the number of decision variables,  $m$  is the number of constraints, and  $N$  is the sample size. The subproblems can be solved very efficiently for problems of approximately  $n + m + N \sim 10^4$ . However, these computations are unacceptable for larger-scale problems and may not even be possible in a single iteration.

The ADMM can be interpreted as a first-order method, in which a method that does not use second-derivative information, such as the Hessian matrix of the objective function. The first-order method has lower approximation accuracy than the second-order method, and thus, a large number of iterations for convergence is necessary when high accuracy is required. However, as the per-iteration solution time to solve the approximation problem is short, this may provide an effective solution for the above-mentioned large-scale problem, unless particularly high accuracy is required.

In particular, in the field of machine learning, several studies, such as Boyd *et al.* [38] and Nedic [59], demonstrated that the computation time could be reduced significantly by dividing the training data into multiple blocks and performing distributed learning. In this study, we expect that large-scale problems can be solved by dividing the training data in the same manner. In particular, as indicated in Table 1, the solution time is short for small problems, and thus, the decomposition method is expected to reduce the computation time significantly.

The contributions of this research are as follows:

- 1) An efficient algorithm for DRO is developed that can be applied to large-scale data. In recent years,

DRO has been applied in many fields including maximum likelihood estimation in machine learning, and it is extremely important to develop a solution for large-scale data.

- 2) The possibility of CO to a new application is demonstrated. Although the concept of CO is not fairly new, it has received considerable attention recently in the light of the big data and cloud computing, and it is important to find new opportunities for applications.
- 3) In the numerical experiments, we present application examples and verify the effect of the proposed method for solving problems of representative classes such as LP, quadratic programming (QP), and SOCP.

The remainder of the paper is organized as follows: In section II, we review the related research. Section III presents the DRO framework. In section IV, the proposed algorithm using CO is presented. The numerical experiments are demonstrated in section V. Section VI provides the conclusions and outlines future challenges.

## II. LITERATURE REVIEW

### A. ROBUST OPTIMIZATION

Robust optimization is a popular approach for optimization under uncertainty. The key concept is to define an uncertainty set of possible realizations of uncertain parameters and subsequently to optimize against worst-case realizations within this set Bertsimas *et al.* [4].

Charnes and Cooper [5] first proposed chance-constrained programming. Soyster [6] presented the concept of uncertainty sets of parameters and determined the solution for the worst-case value. Ben-Tal and Nemirovski [7]–[9] and Ghaoui *et al.* [10], [11] constructed a theoretical foundation for modern robust optimization, with a focus on deriving a tractable robust counterpart for the LP under ellipsoidal parameter uncertainty. Bertsimas and Sim [12] proposed the concept of the “price of robustness,” which flexibly adjusted the level of conservatism of the robust solutions in terms of the probabilistic bounds of constraint violations.

Extensive review papers are available on the subject; see Ben-Tal *et al.* [13], [14], Gorissen *et al.* [15], Gabrel *et al.* [16], Sozuer and Thiele [17], Delage and Iancu [18], and the references therein.

Recent studies have been conducted in connection with stochastic programming. Bandi and Bertsimas [19] proposed a novel approach to analyze stochastic systems based on robust optimization in order to overcome the computational intractability with high dimensions. Nemirovski [20] presented several simulation-based and simulation-free computationally tractable approximations of chance-constrained convex programs, primarily those of chance-constrained linear, conic quadratic, and semidefinite programming. Ben-Tal *et al.* [21] proposed a systematic means of constructing the robust counterpart of a nonlinear uncertain inequality that was concave in the uncertain parameters, using support functions, conjugate functions, and Fenchel duality.

### B. DISTRIBUTIONALLY ROBUST OPTIMIZATION

DRO is a paradigm for decision making under uncertainty whereby the uncertain problem data are governed by a probability distribution that is itself subject to uncertainty. Delage and Ye [22] proposed a DRO model with a moment-based ambiguity set. Ben-Tal *et al.* [23] studied the problem of constructing robust classifiers when the training was plagued with uncertainty. They employed Bernstein bounding schemes to relax the chance-constrained problem as convex second-order cone programming (SOCP), the solution of which was guaranteed to satisfy the probabilistic constraint. Dupacova and Kopa [24] investigated the robustness for stochastic programs in which the set of feasible solutions was dependent on the unknown probability distribution  $P$ , and they derived local bounds using a contamination technique. Xu *et al.* [25] studied probabilistic interpretations of robust optimization. They established a connection between robust optimization and DRO, demonstrating that the solution to any optimization problem is also a solution to a DRO problem. They considered the case in which multiple uncertain parameters belonged to the same fixed dimensional space and determined the set of distributions of the equivalent DRO problem. Zymler *et al.* [26] developed tractable semidefinite programming-based approximations for distributionally robust individual and joint chance constraints, assuming that only the first- and second-order moments as well as the support of the uncertain parameters were provided. Sun *et al.* [27] developed a distributionally robust joint chance-constrained optimization model for a dynamic network design problem under demand uncertainty. Wiesemann *et al.* [30] introduced standardized ambiguity sets that contained all distributions with prescribed conic representable confidence sets and with mean values residing on an affine manifold. They derived conditions under which DRO problems based on standardized ambiguity sets were computationally tractable. Ben-Tal *et al.* [21] proposed robust linear optimization problems with uncertainty regions defined by  $\phi$ -divergences. Bertsimas and Kallus [2] proposed the concept of “predictive prescription.” Within this framework, the objective was to minimize the conditional expected cost wherein a decision was selected in an optimal manner to minimize an uncertain cost that depended on a random variable based on an observation of auxiliary covariates. Bertsimas and Van Parys [29] proposed a framework known as “bootstrap robust analytics,” which integrated DRO and the statistical bootstrap that were designed to produce out-of-sample guarantees by exploiting the use of a confidence region, derived from  $\phi$ -divergence. Esfahani and Kuhn [3] proposed an ambiguity set that was derived from the Wasserstein distance. Rahimian and Mehrotra [32] reviewed the recent advancements in DRO from the modeling, theoretical, and algorithmic perspectives. Kirschner *et al.* [33] studied the DRO problem with an unknown objective function, namely distributionally robust Bayesian optimization. They proposed an algorithm with a performance guarantee, in which the performance was measured by the kernel-based maximum mean discrepancy dis-

tance. Chen *et al.* [34] proposed a decomposition algorithm for two-stage DRO with a phi-divergence ambiguity set. They introduced the nonanticipativity constraints for the first-stage decision and decomposed the problem using Lagrangian relaxation. Huang *et al.* [35] studied multi-stage DRO with the coherence-risk measure. They proposed decomposition methods based on the cutting-plane method. Moreover, they applied their proposed algorithms to the multi-product assembly and portfolio problems, and demonstrated that the risk-averse approach outperformed the risk-neutral approach.

As mentioned above, various modeling methods for uncertainties have been proposed, leading to an equivalent convex robust counterpart. However, in many cases, the robust counterpart is a nonlinear problem, and therefore, the calculation time may be very long. In particular, it is difficult to apply such methods to datasets in which the numbers of samples and dimensions are large. Therefore, it is very important to develop an algorithm that can solve large-scale data efficiently.

### C. ALTERNATING DIRECTION METHOD OF MULTIPLIERS

CO is a type of decomposition method and its decomposition principle is based on the ADMM. The ADMM was proposed by Glowinski and Marrocco [36] and further developed by Gabay and Mercier [37]. For details on the ADMM, refer to Boyd *et al.* [38]. The ADMM is a form of the operator splitting method, which was proposed in the 1950s. The ADMM has been proven to be equivalent to various operator splitting methods. Glowinski [39] first derived the convergence rate of the ADMM. Han and Yuan [40], and Davis and Yin [41], [42] derived the convergence rate of the ADMM in different settings. Makhdomi and Ozdaglar [43] demonstrated the convergence rate of a distributed ADMM over networks. Liu *et al.* [44] proposed a communication-censored ADMM for decentralized CO. Falsone *et al.* [45] proposed the tracking ADMM for distributed constraint-coupled optimization. Eisen *et al.* [46] presented a primal-dual quasi-Newton method for exact CO. In recent years, many examples of the application to machine learning have been provided [47]–[55]). Pinnau *et al.* [56] proposed an integrated framework of swarm intelligence metaheuristics and CO. They presented a gradient-free optimization method for general non-convex programs. Xu *et al.* [57] proposed consensus ADMM in which each agent automatically tuned the local penalty parameters in an adaptive manner. Fang *et al.* [58] proposed a consensus ADMM approach whereby the Newton method was applied for each subproblem to improve the quality of the subproblem solutions. They applied their proposed method to multi-class classification problems and demonstrated the superior performance over other state-of-the-art methods. Carrillo *et al.* [60] improved the algorithm of Pinnau *et al.* [56] by introducing component-wise isotropic Brownian motion and the random selection of mini-batches. Chen *et al.* [61] considered a CO problem with data edge computing in which there were communication bottlenecks and nodes with slow responses. They proposed a coded

stochastic incremental ADMM to mitigate the impact of the straggling nodes by leveraging the data redundancy. They demonstrated that the proposed algorithm had a convergence rate of  $O(1/\sqrt{k})$ . Chen *et al.* [62] presented a randomized incremental primal dual method to solve the CO problem, whereby the dual variable over the connected multi-agent network in each iteration was only updated at a randomly selected node.

## III. DISTRIBUTIONALLY ROBUST OPTIMIZATION

### A. DATA-DRIVEN STOCHASTIC PROGRAMMING

We consider the following stochastic programming problem:

$$\begin{aligned} & \text{minimize } \mathbb{E}f(x, u) \\ & \text{subject to } x \in X, \end{aligned} \quad (1)$$

where  $x \in \mathbb{R}^n$  is the decision variable,  $u \in \mathbb{R}^r$  is the unknown parameter,  $f(x, u)$  is the objective function, and  $X$  is the feasible set. In practice, the distribution  $p(u)$  is not known, and it must therefore be inferred from the data. These are known as *data-driven settings*. In the data-driven settings,  $p(u)$  is partially observable through a finite set of  $M$  independent samples, *e.g.*, past realization of the random variable

$$\mathcal{U}_T := \{u_1, \dots, u_M\},$$

which is known as the *training dataset*. In the training phase, a decision  $\hat{x}_T$  is sought by solving the following problem (2):

$$\begin{aligned} & \text{minimize } \mathbb{E}[f(x, u)|\mathcal{U}_T] \\ & \text{subject to } x \in X. \end{aligned} \quad (2)$$

This solution  $\hat{x}_T$  is known as the *data-driven solution* and its objective function value  $\hat{z}_T = \mathbb{E}[f(\hat{x}_T, u)|\mathcal{U}_T]$  is known as the *certificate*.

The goal of a data-driven problem is to minimize the *out-of-sample* performance of a data-driven solution  $\hat{x}_T$  that is defined as (3).

$$\hat{z}_V = \mathbb{E}f(\hat{x}_T, u). \quad (3)$$

However, as  $p(u)$  is unknown, the exact out-of-sample performance cannot be evaluated in practice; therefore, it is evaluated by the *validation dataset*  $\mathcal{U}_V = \{\hat{u}_1, \dots, \hat{u}_N\}$ , as follows (4):

$$\mathbb{E}_p f(\hat{x}_T, u) \simeq \frac{1}{N} \sum_{j=1}^N f(\hat{x}_T, \hat{u}_j). \quad (4)$$

A natural approach for generating data-driven solutions  $\hat{x}_T$  is the SAA formulation that approximate  $d(p)$  with the empirical distribution  $\mathbf{prob}(u = u_j) = p_j = (1/N)$ . The SAA formulation with training samples  $u_j$  can be expressed as (5)

$$\begin{aligned} & \text{minimize } \frac{1}{M} \sum_{j=1}^M f(x, u_j) \\ & \text{subject to } x \in X. \end{aligned} \quad (5)$$

However, this formulation often leads to poor out-of-sample performance.

**B. DISTRIBUTIONALLY ROBUST OPTIMIZATION**

In this section, we present the DRO problem. DRO is a powerful paradigm for solving the data-driven stochastic programming problem (1). DRO has the following form:

$$\begin{aligned} & \text{minimize } \sup_{p \in \mathcal{P}} \mathbb{E}_{p(u)} f(x, u) \\ & \text{subject to } x \in X, \end{aligned} \tag{6}$$

where  $\mathcal{P}$  is the *ambiguity set* of the probability distribution, which is a family of probability distribution. In DRO, the worst-case expected cost is minimized, whereby the expectation is taken over the ambiguity set.

The ambiguity set  $\mathcal{P}$  is the fundamental input of DRO, and it is desirable to have the following properties:  $\mathcal{P}$  should be sufficiently rich to contain the true data-generating distribution with high confidence;  $\mathcal{P}$  should be sufficiently small to exclude pathological distributions, which would incentivize overly conservative decisions;  $\mathcal{P}$  should also be easy to parameterize from the data; and  $\mathcal{P}$  should facilitate a tractable reformulation of the DRO problem.

Several methods are available to form the ambiguity set  $\mathcal{P}$  from the data, including moment ambiguity sets, confidence regions of goodness-of-fit tests, a ball in the space of the probability distributions using a probability distance function such as the Prohorov metric and Wasserstein metric, and  $\phi$ -divergence. This study examines the properties of DRO problems in which the distributional uncertainty is handled via  $\phi$ -divergences Ben-Tal *et al.* [28].

$\phi$ -divergences measure the distance between two nonnegative vectors  $p = (p_1, \dots, p_M)^T$  and  $q = (q_1, \dots, q_M)^T$ , where  $p$  and  $q$  satisfy  $\sum_{j=1}^M p_j = \sum_{j=1}^M q_j = 1$ . The  $\phi$ -divergence is defined as

$$D(p, q) = \sum_{j=1}^M q_j \phi\left(\frac{p_j}{q_j}\right), \tag{7}$$

where  $\phi(t)$ , which is known as the  $\phi$ -divergence function, is a convex function on  $t \geq 0$ . The  $\phi$ -divergence satisfies  $D(p, q) \geq 0$  and  $D(p, q) = 0$  if and only if  $p = q$ , and it can thus be used as a measure of deviation between two positive vectors.

Using the  $\phi$ -divergence, the ambiguity set  $\mathcal{P}$  can be expressed as (8):

$$\mathcal{P} = \{p : D(p, q) \leq \eta, \sum_{j=1}^M p_j = 1, p_j \geq 0, \forall j\}, \tag{8}$$

where  $q$  is the nominal value with  $q_j = 1/M$  and  $\eta$  is the target distance. By restricting the probability distribution in the ambiguity set, *i.e.*,  $p \in \mathcal{P}$ , the optimization model hedges against distributional uncertainty. When  $\eta$  is set appropriately, the decision maker can control the risk preferences between the risk-neutral and risk-averse approaches.

**C. FORMULATION**

The formulation of DRO with  $\phi$ -divergence is as follows (9):

$$\begin{aligned} & \text{minimize } \sup_p \sum_{j=1}^M p_j f(x, u_j) \\ & \text{subject to } x \in X \\ & \quad D(p, q) \leq \rho, \\ & \quad \sum_{j=1}^M p_j = 1 \\ & \quad p \geq 0. \end{aligned} \tag{9}$$

We present the dual formulation to derive the closed form of the inner maximization. For a given  $x$ , the inner maximization is a convex optimization problem. The inner problem is formulated as (10)

$$\begin{aligned} & \text{maximize}_p \sum_{j=1}^M p_j f(x, u_j) \\ & \text{subject to } D(p, q) \leq \rho, \\ & \quad \sum_{j=1}^M p_j = 1, \\ & \quad p_j \geq 0. \end{aligned} \tag{10}$$

Let  $\lambda$  and  $\mu$  denote the Lagrangian multipliers. When the first and second constraints are multiplied by  $\lambda$  and  $\mu$ , and the constraints are eliminated, we obtain the Lagrangian (11)

$$\begin{aligned} \mathcal{L}(p, \mu, \lambda) = & \sum_{j=1}^M p_j f(x, u_j) + \lambda \rho - \lambda \sum_{j=1}^M q_j \phi\left(\frac{p_j}{q_j}\right) \\ & + \mu - \mu \sum_{j=1}^M p_j. \end{aligned} \tag{11}$$

For simplicity of exposition, we use  $s_j$  to denote the following:

$$s_j = \frac{f(u_j, x) - \mu}{\lambda} \Leftrightarrow f(x, u_j) = \lambda s_j + \mu.$$

Using this expression, we obtain the following reformulation:

$$\begin{aligned} \mathcal{L}(p, \mu, \lambda) = & \sum_{j=1}^M p_j (\lambda s_j + \mu) + \lambda \rho - \lambda \sum_{j=1}^M q_j \phi\left(\frac{p_j}{q_j}\right) \\ & + \mu - \mu \sum_{j=1}^M p_j \\ = & \lambda \rho + \mu + \lambda \sum_{j=1}^M (p_j s_j - q_j \phi\left(\frac{p_j}{q_j}\right)) \\ = & \lambda \rho + \mu + \lambda \sum_{j=1}^M q_j \left(s_j \frac{p_j}{q_j} - \phi\left(\frac{p_j}{q_j}\right)\right). \end{aligned}$$

According to the definition of the conjugate function

$$\phi^*(s) = \sup_{t \geq 0} \{st - \phi(t)\},$$

we obtain the following reformulation:

$$\begin{aligned} & \lambda\rho + \mu + \sup_{p \geq 0} \lambda \sum_{j=1}^M q_j(s_j \frac{p_j}{q_j} - \phi(\frac{p_j}{q_j})) \\ & = \lambda\rho + \mu + \lambda \sum_{j=1}^M q_j \phi^*(s_j). \end{aligned}$$

Combining the dual problem with the outer minimization results in the dual formulation (12)

$$\begin{aligned} & \text{minimize } \mu + \rho\lambda + \lambda \sum_{j=1}^M q_j \phi^*(s_j) \\ & \text{subject to } f(x, u_j) = \lambda s_j + \mu, \quad j = 1, \dots, N \\ & \quad x \in X \\ & \quad \lambda \geq 0. \end{aligned} \quad (12)$$

For further details, refer to Ben-Tal *et al.* [28] and Bayrak- san and Love [31].

#### IV. CONSENSUS OPTIMIZATION

DRO becomes more difficult to apply as the sample size increases. However, the calculation time is short when using the decomposition method and it becomes possible to solve even large-scale problems. In this research, we propose a distributed optimization algorithm using CO, which has attracted substantial attention in the field of convex optimization in recent years.

##### A. PROPOSED ALGORITHM

The central concept of the proposed algorithm is that the problem can be decomposed into training data blocks. As each subproblem results in different solutions  $\hat{x}_T^k$ , a consensus constraint is imposed that ensures that these solutions agree, *i.e.* are equal.

We divide the training data into  $K$  blocks and the problem into  $K$  subproblems. The  $k$ -th subproblem is expressed as (13)

$$\begin{aligned} & \text{minimize } \mu_k + \rho\lambda_k + \lambda_k \sum_{j=1}^M q_j \phi^*(s_j) \\ & \text{subject to } f(x_k, u_j) = \lambda_k s_j + \mu_k, \quad j \in \mathcal{J}_k \\ & \quad x_k \in X \\ & \quad \lambda_k \geq 0, \end{aligned} \quad (13)$$

where  $x_k$ ,  $\mu_k$ , and  $\lambda_k$  are the decision variables for the  $k$ -th problem and  $\mathcal{J}_k$  is the index set for the  $k$ -th block.

Using this subproblem, the original problem can be formulated as (14)

$$\begin{aligned} & \text{minimize } \sum_{k=1}^K F_k(\tilde{x}_k) \\ & \text{subject to } \tilde{x}_k - z = 0, \quad k = 1, \dots, K, \end{aligned} \quad (14)$$

where  $F_k(\tilde{x})$  is the optimal value of DRO for the  $k$ -th block,  $\tilde{x} = [x^T, \mu, \lambda]^T$  are local variables, and  $z$  is a global variable. The equality constraint is a consensus constraint that indicates that the local variables are equal.

It has been established that this CO can be solved efficiently by applying the ADMM. The augmented Lagrangian is expressed as (15)

$$L_\rho(\tilde{x}, z, y) = \sum_{k=1}^K \left( F_k(\tilde{x}_k) + y_k^T (\tilde{x}_k - z) + (\rho/2) \|\tilde{x}_k - z\|_2^2 \right), \quad (15)$$

where  $\rho$  is known as the penalty parameter. In the ADMM,  $\tilde{x}$ ,  $y$ ,  $z$  are updated alternately.

$$\begin{aligned} \tilde{x}_k^{t+1} := & \operatorname{argmin}_{\tilde{x}_k} \left( F_k(\tilde{x}_k) + y_k^{tT} (\tilde{x}_k - z^t) \right. \\ & \left. + (2/\rho) \|\tilde{x}_k - z^t\|_2^2 \right) \end{aligned} \quad (16)$$

$$z^{t+1} := \frac{1}{K} \sum_{k=1}^K \left( \tilde{x}_k^{t+1} + (1/\rho) y_k^t \right) \quad (17)$$

$$y_k^{t+1} := y_k^t + \rho(\tilde{x}_k^{t+1} - z^{t+1}). \quad (18)$$

Using  $\sum_{k=1}^K y_k^t = 0$ , as described in (25), the algorithm is expressed as follows:

$$\begin{aligned} \tilde{x}_k^{t+1} := & \operatorname{argmin}_{\tilde{x}_k} \left( F_k(\tilde{x}_k) + y_k^{tT} (\tilde{x}_k - \bar{x}^t) \right. \\ & \left. + (\rho/2) \|\tilde{x}_k - \bar{x}^t\| \right) \end{aligned} \quad (19)$$

$$y_k^{t+1} := y_k^t + \rho(\tilde{x}_k^{t+1} - \bar{x}^{t+1}), \quad (20)$$

where  $\bar{x}^t = (1/K) \sum_{k=1}^K \tilde{x}_k^t$ . At each iteration, the local variable  $\tilde{x}_k^t$  is updated, the average  $\bar{x}^t$  is obtained, and  $y^t$  is updated to reduce the deviation between the local variable and the average.

##### B. OPTIMALITY CONDITION AND STOPPING CRITERIA

The Lagrangian of CO is expressed as (21)

$$L(\tilde{x}, z, y) = \sum_{k=1}^K \left( F_k(\tilde{x}_k) + y_k^T (\tilde{x}_k - z) \right). \quad (21)$$

The optimality condition of the problem is the feasibility of the main problem (22):

$$\tilde{x}_k^* - z^* = 0 \quad (22)$$

and the dual feasibility formulated as (23)(24):

$$\nabla F_k(\tilde{x}_k^*) + y_k^* = 0 \quad (23)$$

$$\sum_{k=1}^K y_k^* = 0. \quad (24)$$

In this case,  $z^{t+1}$  minimizes  $L_\rho(\tilde{x}_k^{t+1}, z, y^t)$ , and the following equation (25) holds:

$$0 = \nabla_z L_\rho(\tilde{x}_k^{t+1}, z, y^t) \Big|_{z=z^{t+1}}$$

$$\begin{aligned}
 &= \sum_{k=1}^K \left( y_k^t + \rho(\tilde{x}_k^{t+1} - z^{t+1}) \right) \\
 &= \sum_{k=1}^K y_k^{t+1}. \tag{25}
 \end{aligned}$$

Therefore,  $\sum_{k=1}^K y_k^t = 0$  always holds, and thus, the equation (24) is always satisfied. Similarly,  $\tilde{x}^{t+1}$  minimizes  $L_\rho(\tilde{x}, z^t, y^t)$ , and the following equation (26) holds:

$$\begin{aligned}
 0 &= \nabla F_k(\tilde{x}_k^{t+1}) + y_k^t + \rho(\tilde{x}_k^{t+1} - \bar{x}^t) \\
 &= \nabla F_k(\tilde{x}_k^{t+1}) + y_k^{t+1} + \rho(\bar{x}^{t+1} - \bar{x}^t). \tag{26}
 \end{aligned}$$

Therefore, the third term can be regarded as a dual residual. The main residual  $r^t$  and dual residual  $s^t$  can be defined as (27)(28)

$$r^t := \sum_{k=1}^K (\tilde{x}_k^t - \bar{x}^t) \tag{27}$$

$$s^t := \rho \sum_{k=1}^K (\bar{x}^t - \bar{x}^{t-1}). \tag{28}$$

The stopping criteria are that the primal residual  $r^t$  and dual residual  $s^t$  fall within the tolerance  $\varepsilon_{pri}, \varepsilon_{dual}$ , which can be described as (29)(30)

$$\|r^t\|_2 \leq \varepsilon_{pri} \tag{29}$$

$$\|s^t\|_2 \leq \varepsilon_{dual}. \tag{30}$$

The convergence proof for the consensus optimization for the general convex problem can be found in several references, such as [43]–[45].

### C. OVERALL ALGORITHM

The overall algorithm is summarized as follows:

---

**given**  $\rho, \varepsilon_{pri}, \varepsilon_{dual}$

**repeat** 1-4

1.  $\tilde{x}$ -update

$$x_k^{t+1} := \operatorname{argmin}_{x_k} (F_k(\tilde{x}_k) + y_k^{tT}(\tilde{x}_k - \bar{x}^t) + (\rho/2)\|\tilde{x}_k - \bar{x}^t\|)$$

2.  $\bar{x}$ -update:  $\bar{x}^{t+1} = \sum_{k=1}^K \tilde{x}_k^{t+1}$

3.  $y$ -update:  $y_k^{t+1} := y_k^t + \rho(\tilde{x}_k^{t+1} - \bar{x}^{t+1})$

4.  $t$ -update:  $t := t + 1$

**until**  $\|r^t\|_2 \leq \varepsilon_{pri}$  and  $\|s^t\|_2 \leq \varepsilon_{dual}$ .

---

The proposed method is efficient because it divides a large-scale problem into multiple small-scale easier subproblems to solve. Moreover, the dual variables are updated so that the difference between  $\tilde{x}_k$  and  $\bar{x}$  is reduced in order for a consensus to be formed.

## V. NUMERICAL EXAMPLES

The results of the numerical experiments are presented in this section. In the experiments, KL divergence was used as the  $\phi$ -divergence function to construct the DRO. The distributionally robust counterparts (DRCs) of these problems were all convex programming problems. An explanation is provided in subsection A.

We randomly generated problem examples for the typical convex programming problems, namely LP, QP, and SOCP. The generation and results of each problem example are presented in subsections B to D. The combination of the number of variables  $n$  and the number of constraints  $m$  was set to  $(n, m) = \{(20, 30), (200, 300), (2000, 3000)\}$  for the problem size. In each case, the sample size  $N$  was increased to  $N = \{10^1, 10^2, \dots\}$ , and if no answer was provided within 2 h, N/A was assigned. The number of block divisions was  $K = 10$ . The intersection of the termination conditions was  $\varepsilon_{pri} = 10^{-1}, \varepsilon_{dual} = 10^{-1}$ .

We compared the results to ECOS [63] and SCS [64], state-of-art interior-point solver and ADMM solver, respectively. All experiments were conducted in an experimental environment using an Intel (R) Core (TM) i7-8700 CPU 3.20 GHz 3.19 GHz with 32 GB of memory. The program was coded by julia, and the ECOS and SCS solver was called using the convex.jl package.

### A. KL DIVERGENCE

Various  $\phi$ -divergence functions have been proposed to date, each of which performs effectively [31]. In this study, we used the KL divergence between  $p, q$ , expressed as (31)

$$D_{kl}(p, q) = \sum_{j=1}^M (p_j \log(p_j/q_j) - p_j + q_j). \tag{31}$$

Using KL divergence, the ambiguity set  $\mathcal{P}$  can be expressed as (32)

$$\mathcal{P} = \{p : \sum_{j=1}^M D_{kl}(p, q) \leq \rho, \sum_{j=1}^M p_j = 1, p_j \geq 0, \forall j\}. \tag{32}$$

The conjugate of the KL divergence is formulated as (33)

$$\phi^*(s) = e^s - 1. \tag{33}$$

The formulation of DRO using KL divergence is expressed as (34)

$$\begin{aligned}
 &\text{minimize } \mu + \rho\lambda + \lambda \sum_{j=1}^M q_j(e^{s_j} - 1) \\
 &\text{subject to } f(x, u_j) = \lambda s_j + \mu, \quad j = 1, \dots, N \\
 &\quad x \in X \\
 &\quad \lambda \geq 0. \tag{34}
 \end{aligned}$$

### B. LINEAR PROGRAMMING

#### 1) PROBLEM INSTANCES

We considered the following LP problem (35):

$$\begin{aligned}
 &\text{minimize } c^T x \\
 &\text{subject to } Ax \geq b. \tag{35}
 \end{aligned}$$

**TABLE 2.** Comparison of computation time of DRO for LP.

$n$	$m$	$N$	ECOS	SCS	CDRO	Iterations
20	30	$10^1$	<b>0.037</b>	0.085	0.399	8
20	30	$10^2$	<b>0.039</b>	1.389	0.521	8
20	30	$10^3$	<b>1.517</b>	46.157	2.662	6
20	30	$10^4$	<b>113.319</b>	335.843	188.398	9
20	30	$10^5$	N/A	N/A	<b>5025.185</b>	20
20	30	$10^6 \geq$	N/A	N/A	N/A	N/A
200	300	$10^1$	<b>0.166</b>	3.019	2.130	2
200	300	$10^2$	<b>0.833</b>	8.271	2.487	2
200	300	$10^3$	<b>11.096</b>	44.026	25.690	6
200	300	$10^4$	754.584	901.414	<b>394.391</b>	10
200	300	$10^5 \geq$	N/A	N/A	N/A	N/A
2000	3000	$10^1 \geq$	N/A	N/A	<b>2261.016</b>	2
2000	3000	$10^2 \geq$	N/A	N/A	N/A	N/A

The problem instance was generated as follows: Each element of  $c$  was randomly generated from the uniform distribution  $\mathcal{U}(0, 1)$ , and its absolute value was obtained. Each element of  $A$  was randomly generated from the normal distribution  $\mathcal{N}(0, 1^2)$  and its absolute value was obtained. Moreover,  $b$  generated a random solution  $x_0$  and sets  $b = Ax_0$ . Each element of  $x_0$  was randomly generated from the normal distribution  $\mathcal{N}(0, 1^2)$  and its absolute value was obtained.

## 2) RESULTS

Table 2 displays the experimental results. The computation time of ECOS, SCS and CDRO are shown, where ECOS and SCS refer to the DRC of problem (35) that was solved by each solver directly, and CDRO refers to the problem optimized by the proposed method. The number of iterations for CDRO is indicated.

According to Table 2, when  $(n, m) = (20, 30)$ , ECOS was faster up to  $N \leq 10^4$ , but when  $N = 10^5$ , the problem could be solved only by CDRO. In the case of  $(n, m) = (200, 300)$ , ECOS was faster until  $N \leq 10^3$ , and CDRO was faster at  $N = 10^4$ . In the case of  $(n, m) = (2000, 3000)$ , when  $N = 10^1$ , the problem could not be solved by ECOS, but it could be solved by CDRO.

The results demonstrate that ECOS is faster when the sample size is small and CDRO is faster when the sample size increases. Moreover, problems that cannot be solved by ECOS or SCS can be solved by CDRO, particularly when the numbers of dimensions and samples are large. The effectiveness of the proposed technique was validated based on these results.

## C. QUADRATIC PROGRAMMING

### 1) PROBLEM INSTANCES

We considered the following QP problem (36):

$$\begin{aligned} & \text{minimize } x^T Qx + c^T x \\ & \text{subject to } lb \leq Ax \leq ub. \end{aligned} \quad (36)$$

The problem instance was generated as follows: Each element of  $c$  was randomly generated from the normal distribution  $\mathcal{N}(0, 1^2)$  and its absolute value was obtained. Furthermore,  $lb$  was generated from the uniform distribution  $-\mathcal{U}(0, 1)$  and  $ub$  was generated from the uniform distribution  $\mathcal{U}(0, 1)$ .

**TABLE 3.** Comparison of computation time of DRO for QP.

$n$	$m$	$N$	ECOS	SCS	CDRO	Iterations
20	30	$10^1$	<b>0.043</b>	0.047	0.404	5
20	30	$10^2$	<b>0.387</b>	0.518	0.468	2
20	30	$10^3$	25.674	37.204	<b>9.258</b>	2
20	30	$10^4$	N/A	N/A	<b>576.929</b>	2
20	30	$10^5 \geq$	N/A	N/A	N/A	N/A
200	300	$10^1$	4.586	4.910	<b>3.830</b>	2
200	300	$10^2$	45.438	80.913	<b>43.451</b>	2
200	300	$10^3$	N/A	N/A	<b>3137.053</b>	5
200	300	$10^4 \geq$	N/A	N/A	N/A	N/A
2000	3000	$10^1 \geq$	N/A	N/A	N/A	N/A

## 2) RESULTS

The experimental results are presented in Table 3. When  $(n, m) = (20, 30)$ , ECOS was faster up to  $N \leq 10^2$ , and CDRO was faster from  $N \geq 10^3$ . In particular, when  $N = 10^4$ , the problem could be solved only by CDRO. In the case of  $(n, m) = (200, 300)$ , CDRO was faster in all cases, and in the case of  $N = 10^4$ , the problem could be solved only by CDRO.

The results demonstrate that CDRO is faster as the problem and sample sizes increase, as in the case of the LP problem. Moreover, CDRO can solve problems that cannot be solved by ECOS or SCS, particularly when the numbers of dimensions and samples are large. The effectiveness of the proposed technique was validated based on these results.

## D. SECOND-ORDER CONE PROGRAMMING

### 1) PROBLEM INSTANCES

We considered the following SOCP problem (37):

$$\begin{aligned} & \text{minimize } f^T x \\ & \text{subject to } \|Ax + b\|_2 \leq c^T x + d. \end{aligned} \quad (37)$$

The problem instance was generated as follows: Each element of  $f, A, b, c$  was randomly generated from the normal distribution  $\mathcal{N}(0, 1^2)$  and its absolute value was obtained. For  $d$ , a random solution  $x_0$  was generated and set as  $d = \|Ax_0 + b\|_2 - c^T x_0$ . Each element of  $x_0$  was randomly generated from the normal distribution  $\mathcal{N}(0, 1^2)$  and its absolute value was obtained.

## 2) RESULTS

The experimental results are displayed in Table 4. When  $(n, m) = (20, 30)$ , ECOS was faster up to  $N \leq 10^3$ , whereas CDRO was faster from  $N \geq 10^4$ . In particular, when  $N = 10^5$ , the problem could be solved only with CDRO. When  $(n, m) = (200, 300)$ , the normal DRO was faster up to  $N \leq 10^3$  and CDRO was faster at  $N = 10^4$ .

The results indicate that CDRO is faster when the problem and sample sizes are large, as in the case of the LP and QP problems. Furthermore, and CDRO can solve large-scale problems that ECOS or SCS cannot solve. The effectiveness of the proposed technique was validated based on these results.



**TABLE 4. Comparison of computation time of DRO for SOCP.**

$n$	$m$	$N$	ECOS	SCS	CDRO	Iterations
20	30	$10^1$	<b>0.031</b>	0.720	0.267	5
20	30	$10^2$	<b>0.032</b>	0.204	0.312	4
20	30	$10^3$	<b>1.386</b>	0.813	2.658	5
20	30	$10^4$	108.865	185.860	<b>70.740</b>	10
20	30	$10^5$	N/A	N/A	<b>6200.178</b>	20
20	30	$10^6 \geq$	N/A	N/A	N/A	N/A
200	300	$10^1$	<b>0.179</b>	1.733	0.753	4
200	300	$10^2$	<b>0.200</b>	23.597	1.809	5
200	300	$10^3$	<b>5.856</b>	5.908	7.423	7
200	300	$10^4$	540.784	453.350	<b>184.126</b>	2
200	300	$10^5 \geq$	N/A	N/A	N/A	N/A
2000	3000	$10^1 \geq$	N/A	N/A	N/A	N/A

**VI. CONCLUSION**

The use of data to make decisions in uncertain situations is becoming increasingly important. A new optimization paradigm known as DRO has been attracting attention as a methodology for this purpose. DRO offers significant potential because it can derive robust decision making from data without assuming a probability distribution. However, DRO problem is difficult to solve and the problem size that can be solved to optimality is limited. In this research, we proposed an efficient algorithm to solve the large-scale DRO problems. We have developed an algorithm that applies CO, which has attracted attentions as a decomposition method for large-scale problems in the convex programming. In CO, a large-scale problem is decomposed into smaller subproblems and a consensus constraint is imposed so that the local solutions of the subproblems become equal. Therefore, the calculation is fast and the solution can be obtained efficiently even in large-scale problems. We conducted numerical experiments in which we applied the proposed method to DRO for LP, QP, and SOCP. The results demonstrate that the proposed method is faster than state-of-art solvers when the problem and sample sizes are large, and the proposed method can solve larger-scale problems. The effectiveness of the proposed method was verified based on these results.

Future issues include research on optimization algorithms for each subproblem. Each subproblem presented in the proposed algorithm is a convex programming. However, even by the state-of-art solver, the numerical stability was not sufficient; thus, the development of an algorithm to solve the subproblem reliably is required. Another expansion is the application of CO to other DRO models. There are other DRO models that use moment ambiguity set and the Wasserstein ambiguity set, and it is expected that the decomposition approach proposed in this research can be used to increase the speed for solving these models. Finally, further speedup can be considered by integrating other decomposition methods. The primal and dual variables of DRO, modeled as one decision variable, can be further decomposed in the ADMM form.

**REFERENCES**

[1] J. E. Smith and R. L. Winkler, "The optimizer's curse: Skepticism and postdecision surprise in decision analysis," *Manage. Sci.*, vol. 52, no. 3, pp. 311–322, Mar. 2006.

[2] D. Bertsimas and N. Kallus, "From predictive to prescriptive analytics," *Manage. Sci.*, vol. 66, no. 3, pp. 1025–1044, Mar. 2020.

[3] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations," *Math. Program.*, vol. 171, nos. 1–2, pp. 115–166, Sep. 2018.

[4] D. Bertsimas, V. Gupta, and N. Kallus, "Data-driven robust optimization," *Math. Program.*, vol. 167, no. 2, pp. 235–292, Feb. 2018.

[5] A. Charnes and W. W. Cooper, "Chance-constrained programming," *Manage. Sci.*, vol. 6, no. 1, pp. 73–79, Oct. 1959.

[6] A. L. Soyster, "Convex programming with set-inclusive constraints and applications to inexact linear programming," *Oper. Res.*, vol. 21, no. 5, pp. 1154–1157, 1973.

[7] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Math. Oper. Res.*, vol. 23, no. 4, pp. 769–805, 1998.

[8] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Oper. Res. Lett.*, vol. 25, no. 1, pp. 1–13, Aug. 1999.

[9] A. Ben-Tal and A. Nemirovski, "Robust solutions of linear programming problems contaminated with uncertain data," *Math. Program.*, vol. 88, no. 3, pp. 411–424, Sep. 2000.

[10] L. El Ghaoui and H. Lebret, "Robust solutions to least-squares problems with uncertain data," *SIAM J. Matrix Anal. Appl.*, vol. 18, no. 4, pp. 1035–1064, Oct. 1997.

[11] L. El Ghaoui, F. Oustry, and H. Lebret, "Robust solutions to uncertain semidefinite programs," *SIAM J. Optim.*, vol. 9, no. 1, pp. 33–52, Jan. 1998.

[12] D. Bertsimas and M. Sim, "The price of robustness," *Oper. Res.*, vol. 52, no. 1, pp. 35–53, Feb. 2004.

[13] A. Ben-Tal and A. Nemirovski, "Selected topics in robust convex optimization," *Math. Program.*, vol. 112, no. 1, pp. 125–158, Jul. 2007.

[14] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, vol. 28. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[15] B. L. Gorissen, İ. Yanıkoğlu, and D. den Hertog, "A practical guide to robust optimization," *Omega*, vol. 53, pp. 124–137, Jun. 2015.

[16] V. Gabrel, C. Murat, and A. Thiele, "Recent advances in robust optimization: An overview," *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 471–483, Jun. 2014.

[17] S. Sozuer and A. C. Thiele, "The state of robust optimization," in *Robustness Analysis in Decision Aiding, Optimization, and Analytics*. Cham, Switzerland: Springer, 2016, pp. 89–112.

[18] E. Delage and D. A. Iancu, "Robust multistage decision making," in *The Operations Research Revolution*. Catonsville, MD, USA: INFORMS, 2015, pp. 20–46.

[19] C. Bandi and D. Bertsimas, "Tractable stochastic analysis in high dimensions via robust optimization," *Math. Program.*, vol. 134, no. 1, pp. 23–70, Aug. 2012.

[20] A. Nemirovski, "On safe tractable approximations of chance constraints," *Eur. J. Oper. Res.*, vol. 219, no. 3, pp. 707–718, Jun. 2012.

[21] A. Ben-Tal, D. den Hertog, and J.-P. Vial, "Deriving robust counterparts of nonlinear uncertain inequalities," *Math. Program.*, vol. 149, nos. 1–2, pp. 265–299, Feb. 2015.

[22] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Oper. Res.*, vol. 58, no. 3, pp. 595–612, Jun. 2010.

[23] A. Ben-Tal, S. Bhadra, C. Bhattacharyya, and J. S. Nath, "Chance constrained uncertain classification via robust optimization," *Math. Program.*, vol. 127, no. 1, pp. 145–173, Mar. 2011.

[24] J. Dupačová and M. Kopa, "Robustness in stochastic programs with risk constraints," *Ann. Oper. Res.*, vol. 200, no. 1, pp. 55–74, Nov. 2012.

[25] H. Xu, C. Caramanis, and S. Mannor, "A distributional interpretation of robust optimization," *Math. Oper. Res.*, vol. 37, no. 1, pp. 95–110, Feb. 2012.

[26] S. Zymler, D. Kuhn, and B. Rustem, "Distributionally robust joint chance constraints with second-order moment information," *Math. Program.*, vol. 137, nos. 1–2, pp. 167–198, Feb. 2013.

[27] H. Sun, Z. Gao, W. Y. Szeto, J. Long, and F. Zhao, "A distributionally robust joint chance constrained optimization model for the dynamic network design problem under demand uncertainty," *Netw. Spatial Econ.*, vol. 14, nos. 3–4, pp. 409–433, Dec. 2014.

[28] A. Ben-Tal, D. den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Manage. Sci.*, vol. 59, no. 2, pp. 341–357, Feb. 2013.

- [29] D. Bertsimas and B. Van Parys, "Bootstrap robust prescriptive analytics," 2017, *arXiv:1711.09974*. [Online]. Available: <http://arxiv.org/abs/1711.09974>
- [30] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally robust convex optimization," *Oper. Res.*, vol. 62, no. 6, pp. 1358–1376, 2014.
- [31] G. Bayraktan and D. K. Love, "Data-driven stochastic programming using phi-divergences," in *The Operations Research Revolution*. Catonsville, MD, USA: INFORMS, 2015, pp. 1–19.
- [32] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," 2019, *arXiv:1908.05659*. [Online]. Available: <http://arxiv.org/abs/1908.05659>
- [33] J. Kirschner, I. J. S. Bogunovic, and A. Krause, "Distributionally robust Bayesian optimization," in *Proc. Int. Conf. Artif. Intell. Statist.*, Jun. 2020, pp. 2174–2184.
- [34] Y. Chen, H. Sun, and H. Xu, "Decomposition and discrete approximation methods for solving two-stage distributionally robust optimization problems," *Comput. Optim. Appl.*, vol. 78, no. 1, pp. 205–238, 2021.
- [35] R. Huang, S. Qu, X. Yang, and Z. Liu, "Multi-stage distributionally robust optimization with risk aversion," *J. Ind. Manage. Optim.*, vol. 17, no. 1, pp. 233–259, 2021.
- [36] R. Glowinski and A. Marrocco, "On the solution of a class of non linear Dirichlet problems by a penalty-duality method and finite elements of order one," in *Proc. Optim. Techn. IFIP Tech. Conf.* Berlin, Germany: Springer, 1975, pp. 327–333.
- [37] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.
- [38] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.
- [39] R. Glowinski, "Splitting methods for the numerical solution of the incompressible Navier–Stokes equations," Madison Math. Res. Center, Wisconsin Univ., Madison, WI, USA, Tech. Rep. MRC-TSR-2741, 1984.
- [40] D. Han and X. Yuan, "A note on the alternating direction method of multipliers," *J. Optim. Theory Appl.*, vol. 155, no. 1, pp. 227–238, Oct. 2012.
- [41] D. Davis and W. Yin, "Faster convergence rates of relaxed peaceman-rachford and ADMM under regularity assumptions," *Math. Oper. Res.*, vol. 42, no. 3, pp. 783–805, Aug. 2017.
- [42] D. Davis and W. Yin, "Convergence rate analysis of several splitting schemes," in *Splitting Methods in Communication, Imaging, Science, and Engineering*. Cham, Switzerland: Springer, 2016, pp. 115–163.
- [43] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed ADMM over networks," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5082–5095, Oct. 2017.
- [44] Y. Liu, W. Xu, G. Wu, Z. Tian, and Q. Ling, "Communication-censored ADMM for decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2565–2579, May 2019.
- [45] A. Falsone, I. Notariccola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, Jul. 2020, Art. no. 108962.
- [46] M. Eisen, A. Mokhtari, and A. Ribeiro, "A primal-dual quasi-Newton method for exact consensus optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 23, pp. 5983–5997, Dec. 2019.
- [47] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An ADMM algorithm for a class of total variation regularized estimation problems," *IFAC Proc. Volumes*, vol. 45, no. 16, pp. 83–88, Jul. 2012.
- [48] H. Sedghi, A. Anandkumar, and E. Jonckheere, "Multi-step stochastic ADMM in high dimensions: Applications to sparse optimization and matrix decomposition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2771–2779.
- [49] H. Wang and A. Banerjee, "Online alternating direction method (longer version)," 2013, *arXiv:1306.3721*. [Online]. Available: <http://arxiv.org/abs/1306.3721>
- [50] C. Zhang, H. Lee, and K. Shin, "Efficient distributed linear classification algorithms via the alternating direction method of multipliers," in *Proc. Artif. Intell. Statist.*, Mar. 2012, pp. 1398–1406.
- [51] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2014, pp. 1701–1709.
- [52] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "Basis pursuit in sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 2916–2919.
- [53] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.
- [54] N. S. Aybat and G. Iyengar, "An alternating direction method with increasing penalty for stable principal component pursuit," *Comput. Optim. Appl.*, vol. 61, no. 3, pp. 635–668, Jul. 2015.
- [55] N. S. Aybat, S. Zarmehri, and S. Kumara, "An ADMM algorithm for clustering partially observed networks," in *Proc. SIAM Int. Conf. Data Mining*, Jun. 2015, pp. 460–468.
- [56] R. Pinnau, C. Totzeck, O. Tse, and S. Martin, "A consensus-based model for global optimization and its mean-field limit," *Math. Models Methods Appl. Sci.*, vol. 27, no. 1, pp. 183–204, Jan. 2017.
- [57] Z. Xu, G. Taylor, H. Li, M. A. Figueiredo, X. Yuan, and T. Goldstein, "Adaptive consensus ADMM for distributed optimization," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 3841–3850.
- [58] C.-H. Fang, S. B. Kylasa, F. Roosta, M. W. Mahoney, and A. Grama, "Newton-ADMM: A distributed GPU-accelerated optimizer for multiclass classification problems," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. (SC)*, Nov. 2020, pp. 1–12.
- [59] A. Nedic, "Distributed gradient methods for convex machine learning problems in networks: Distributed optimization," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 92–101, May 2020.
- [60] J. A. Carrillo, S. Jin, L. Li, and Y. Zhu, "A consensus-based global optimization method for high dimensional machine learning problems," *ESAIM, Control, Optim. Calculus Variat.*, vol. 27, p. S5, Jul. 2021.
- [61] H. Chen, Y. Ye, M. Xiao, M. Skoglund, and H. V. Poor, "Coded stochastic ADMM for decentralized consensus optimization with edge computing," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5360–5373, Apr. 2021.
- [62] C. Chen, Y. Chen, and X. Ye, "A randomized incremental primal-dual method for decentralized consensus optimization," *Anal. Appl.*, vol. 19, no. 3, pp. 465–489, May 2021.
- [63] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 3071–3076.
- [64] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *J. Optim. Theory Appl.*, vol. 169, no. 3, pp. 1042–1068, Jun. 2016.



**SHUNICHI OHMORI** received the master's and Ph.D. degrees in engineering from Waseda University. He is currently an Associate Professor with the Department of Business Design and Management, Waseda University, Japan, where he is also a Researcher with the Institute of Global Production and Logistics and the Data Science Institute. His research interests include operations research and supply chain management.

...