Coordinate Linear Variance Reduction for Generalized Linear Programming

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study a class of generalized linear programs (GLP) in a large-scale setting, which includes a simple, possibly nonsmooth convex regularizer and simple convex set constraints. By reformulating (GLP) as an equivalent convex-concave min-max problem, we show that the linear structure in the problem can be used to design an efficient, scalable first-order algorithm, to which we give the name *Coordinate Linear Variance Reduction* (CLVR; pronounced "clever"). CLVR yields improved complexity results for (GLP) that depend on the max row norm of the linear constraint matrix in (GLP) rather than the spectral norm. When the regularization terms and constraints are separable, CLVR admits an efficient lazy update strategy that makes its complexity bounds scale with the number of nonzero elements of the linear constraint matrix in (GLP) rather than the matrix dimensions. Further, for the special case of linear programs and by exploiting sharpness, we propose a restart scheme for CLVR to obtain empirical linear convergence. Finally, we show that Distributionally Robust Optimization (DRO) problems with ambiguity sets based on both f-divergence and Wasserstein metrics can be reformulated as (GLPs) by introducing sparsely connected auxiliary variables. We complement our theoretical guarantees with numerical experiments that verify our algorithm's practical effectiveness in terms of wall-clock time and number of data passes.

1 Introduction

We study the following generalized linear program (GLP):

$$\min_{\boldsymbol{x}} \ \left\{\boldsymbol{c}^T\boldsymbol{x} + r(\boldsymbol{x}) : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in \mathcal{X}\right\}, \tag{GLP}$$

where $\boldsymbol{x}, \boldsymbol{c} \in \mathbb{R}^d$, $\boldsymbol{A} \in \mathbb{R}^{n\times d}$, $\boldsymbol{b} \in \mathbb{R}^n$, $r : \mathbb{R}^d \to \mathbb{R}$ is a convex regularizer, and $\mathcal{X} \subseteq \mathbb{R}^d$ is a closed convex set, such that a proximal / projection operator involving r and $\mathcal{X}$ can be computed efficiently. When $\mathcal{X}$ is the nonnegative orthant $\{\boldsymbol{x} : x_i \geq 0, i \in [d]\}$ and $r \equiv 0$, (GLP) reduces to the standard form of a linear program (LP). When $\mathcal{X}$ is a convex cone and $r \equiv 0$, (GLP) reduces to a conic linear program. (GLP) has had a significant impact in traditional engineering disciplines such as transportation, energy, telecommunications, and manufacturing. In modern data science, we observe the renaissance of (GLP) due to its modeling power in such areas as reinforcement learning [17], optimal transport [55], and neural network verification [37]. For traditional engineering disciplines with moderate scale or exploitable sparsity, off-the-shelf interior point methods that form and factorize matrices in each iteration are often good choices as practical solvers [23]. In data science applications, however, where the data are often dense or of extreme scale, the amount of computation and/or memory required by matrix factorization is prohibitive. Thus, first-order methods that avoid matrix factorizations are potentially appealing options. In this context, because the presence of the linear equality constraint in (GLP) may complicate projection operations onto the feasible set, we consider

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

an equivalent reformulation of (GLP) as a min-max problem involving the Lagrangian:

$$\min_{\boldsymbol{x}\in\mathcal{X}\subset\mathbb{R}^d} \max_{\boldsymbol{y}\in\mathbb{R}^n} \Big\{\mathcal{L}(\boldsymbol{x},\boldsymbol{y}) := \boldsymbol{c}^T\boldsymbol{x} + r(\boldsymbol{x}) + \boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}^T\boldsymbol{b}\Big\}. \tag{PD-GLP}$$

In data science applications, both n and d can be very large. (PD-GLP) can be viewed as a structured bilinearly coupled min-max problem, where the linearity of $\mathcal{L}(\boldsymbol{x},\boldsymbol{y})$ in the dual variable vector $\boldsymbol{y}$ is vital to our algorithmic development.

1.1 Background

While the literature addressing (PD-GLP) is sparse — some special cases have been considered in [39–41] — there has been significant recent work on first-order methods for general bilinearly coupled convex-concave min-max problems. Deterministic first-order methods include the proximal point method (PPM) [49], the extragradient/mirror-prox method (EGM) [32, 43], the primal-dual hybrid gradient (PDHG) method [13], and the alternating direction method of multipliers (ADMM) [18]. All these methods have per-iteration cost $\Theta(\mathrm{nnz}(\boldsymbol{A}) + n + d)$ and convergence rate $1/k$, where $\mathrm{nnz}(\boldsymbol{A})$ denotes the number of nonzero elements of $\boldsymbol{A}$ and k is the number of iterations.

For better scalability, stochastic counterparts of these methods have been proposed. [10, 31, 45, 47] have used "vanilla" stochastic gradients to replace the full gradients of their deterministic counterparts. [2, 12, 25] have exploited the finite-sum structure of the interaction term $\langle \boldsymbol{y}, \boldsymbol{A}\boldsymbol{x}\rangle$ involving both primal and dual variables to perform variance reduction. With a separability assumption for the dual variables, [3] and [14] have combined incremental coordinate approaches on the dual variables with an implicit variance reduction strategy on the primal variables. Recently, under a separability assumption for dual variables, [53] have proposed a new incremental coordinate method with an initialization step that requires a single access to the full data. This approach, known as *variance reduction via primal-dual accelerated dual averaging* (VRPDA^2), obtains the first theoretical bounds that are better than their deterministic counterparts in the class of incremental coordinate approaches. The VRPDA^2 algorithm serves as the main motivation for our approach.

It is of particular interest to design algorithms that scale with the number of nonzero elements in $\boldsymbol{A}$ for at least two reasons: (i) the data matrix can be sparse; and (ii) when we consider simplified reformulations of certain complicated models, we often need to introduce sparsely connected auxiliary variables. Nevertheless, the randomized coordinate algorithms of [3, 14, 53] have $O(d)$ per-iteration cost regardless of the sparsity of $\boldsymbol{A}$. To address this issue, [22, 33] have proposed incremental primal-dual coordinate methods with per-iteration cost that scales with the number of nonzero elements in the row from $\boldsymbol{A}$ that is used in each iteration, at the price of smaller step size for dense $\boldsymbol{A}$. Moreover, [5] has proposed a random extrapolation approach that admits both low per-iteration cost and large step size. Despite these developments, all these algorithms produce less accurate iterates than the methods with $O(d)$ per-iteration cost, thus degrading their worst-case complexity.

Finally, for the special case of LP, based on the positive Hoffman constant [28], [9] proved that the primal-dual formulation of LP exhibits a sharpness property, which bounds below the growth of a normalized primal-dual gap from the same work. Based on this sharpness property, [9] proposed a restart scheme for the deterministic first-order methods discussed above to obtain linear convergence. [8] further extended this restart strategy using various heuristics to improve practical performance.

1.2 Motivation

We sharpen the focus from general bilinearly coupled convex-concave min-max problems to (GLP) and its primal-dual formulation (PD-GLP), because many complicated models can be reformulated as (GLP) and (GLP) possesses additional structure that can be exploited in algorithm design. Our motivation for focusing on (GLP) is to bridge the large gap between the well-studied stochastic variance reduced first-order methods [6, 30, 52, 53] and the increasingly popular and complicated, yet highly structured large-scale problems arising in distributionally robust optimization (DRO) [19–21, 29, 36, 42, 50, 56, 58]; see also a recent survey by [48] and references therein.

For DRO problems with ambiguity sets defined by f-divergence [29, 35, 42], the original formulation is a nonbilinearly coupled convex-concave min-max problem. Even the well constructed reformulation in [35] does not admit unbiased stochastic gradients, leading to complicated algorithms and analysis. For DRO problems with ambiguity sets defined by Wasserstein metric [21, 27, 36, 50, 58], the original

formulation is in general infinite-dimensional. (Finite-dimensional reformulations [21, 50] exist for special cases of logistic regression and smooth convex losses.) Solvers that have been proposed for DRO with Wasserstein metric are either multiple-loop deterministic ADMM [36] or are designed for general convex-concave problems [58].

By introducing auxiliary variables with sparse connections,[1] we show that DRO with ambiguity sets based on both f-divergence and the Wasserstein metric can be reformulated as (GLP). Thus, complicated DRO problems can be addressed by a simple, efficient, and scalable algorithm for (GLP). Our algorithm for solving (GLP) and the proposed reformulations of DRO are our main contributions.

1.3 Contributions

Algorithm. Motivated by VRPDA^2 [53], we propose a simple, efficient, and scalable algorithm for (PD-GLP). Our algorithm combines an incremental *coordinate* method with exploitation of the *linear* structure for the dual variables in (PD-GLP) and the implicit *variance reduction* effect in the algorithm, so we name it *coordinate linear variance reduction* (CLVR, pronounced "clever"). CLVR is inspired by VRPDA^2 but customized to the particular structure of (PD-GLP). In particular, by exploiting the fact that the max problem is linear and unconstrained in the dual variable vector $\boldsymbol{y} \in \mathbb{R}^n$, we find that the expensive initialization step used in VRPDA^2 is not needed and we can take simpler and larger steps. Further, in the structured case in which $\boldsymbol{A}$ is sparse and the convex constraint set $\mathcal{X}$ and the regularizer $r(\boldsymbol{x})$ are fully separable[2], we show that the dual averaging update in CLVR enables us to design an efficient lazy update strategy for which the per-iteration cost of CLVR scales with the number of nonzero elements of the selected row from $\boldsymbol{A}$ in each iteration, which is potentially much lower than the order-d cost in VRPDA^2. Finally, CLVR uses extrapolation on dual variables rather than on primal variables considered in VRPDA^2, which significantly reduces implementation complexity of our lazy update strategy for structured variants of (PD-GLP). On the technical side, although both CLVR and VRPDA^2 are randomized algorithms that bound the primal-dual gap in expectation, the guarantee provided by CLVR is stronger as it allows bounding the expectation of the supremum gap as opposed to the supremum of expected gap in VRPDA^2.

To state our complexity results, we make the following scaling assumption.

Assumption 1. $L := \|\boldsymbol{A}\|$ *and each row of* $\boldsymbol{A}$ *in* (GLP) *is normalized with Euclidean norm* R.

As a comment, it is known that preprocessing in modern LP solvers [23] often ensures normalized rows/columns for the data matrix. Observe that $R \leq L \leq \sqrt{n}R$. The upper bound is achieved when all elements of $\boldsymbol{A}$ have identical value. Although this is an extreme case, there exist ill-conditioned practical datasets where we can expect significant performance gains if the complexity can be reduced from $O(L)$ to $O(R)$. In Table 1, we give the overall complexity bounds (total number of arithmetic operations) and the per-iteration cost of a representative set of existing algorithms and our CLVR algorithm for solving structured (PD-GLP), i.e., $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ with $\mathcal{X}_i \in \mathbb{R}\,(i \in [d])$ and $r(\boldsymbol{x}) := \sum_{i=1}^d r(x^i)$. We further assume that for the stochastic algorithms [2, 14, 53] and our CLVR algorithm, we draw one row of $\boldsymbol{A}$ per iteration uniformly at random to make the complexity results comparable. The general convex setting corresponds to $r(\boldsymbol{x})$ being general convex ($\sigma = 0$), while the strongly convex setting corresponds to $r(\boldsymbol{x})$ being σ-strongly convex ($\sigma > 0$).

As shown in Table 1, all the algorithms have the optimal dependence on ϵ [46], while the dependence on the ambient dimensions n, d, the number of nonzero elements of $\boldsymbol{A}$, $\mathrm{nnz}(\boldsymbol{A})$, and the constants L and R are quite different. For both the general convex and strongly convex settings, CLVR is the first algorithm that has no explicit dependence on the ambient dimensions d and n, instead depending on the number of nonzero elements $\mathrm{nnz}(\boldsymbol{A})$. Moreover, the complexity of CLVR depends on the max row norm R rather than the spectral norm L, and the per-iteration cost of CLVR only depends on the nonzero elements of the selected row from $\boldsymbol{A}$ in each iteration, which can be far less than d.

By exploiting the linear structure again, we provide explicit guarantees for both the objective value and the constraint satisfaction of (GLP). Further, the analysis of CLVR applies to the more general block-coordinate update setting, which is better suited to modern parallel computing platforms.

[1] "Sparse connections" here means that even though the newly introduced variables may substantially increase the problem dimensions, the number of nonzero entries in the constraint matrix remains of the same order.

[2] We state the results here for the fully separable setting for convenience of comparison; however, our results are also applicable to the block separable setting.

Table 1: Overall complexity and per-iteration cost for solving structured (PD-GLP). ("—" indicates that the corresponding result does not exist or is unknown.)

Algorithm	General Convex (Primal-Dual Gap)	Strongly Convex (Distance to Solution)	Per-Iteration Cost
PDHG CP(2011)	$O\big(\frac{(\mathrm{nnz}(\boldsymbol{A})+n+d)L}{\epsilon}\big)$	$O\big(\frac{(\mathrm{nnz}(\boldsymbol{A})+n+d)L}{\sigma\sqrt{\epsilon}}\big)$	$O(\mathrm{nnz}(A)+n+d)$
SPDHG CERS(2018)	$O\big(\frac{ndL}{\epsilon}\big)$	$O\big(\frac{ndL}{\sigma\sqrt{\epsilon}}\big)$	$O(d)$
EVR AM (2021)	$O\big(\mathrm{nnz}(\boldsymbol{A})+\frac{\sqrt{\mathrm{nnz}(\boldsymbol{A})(n+d)nR}}{\epsilon}\big)$	—	$O(n+d)$
VRPDA^2 SWD(2021)	$O(nd\log\min\{\frac{1}{\epsilon},n\}+\frac{ndR}{\epsilon})$	$O(nd\log\min\{\frac{1}{\epsilon},n\}+\frac{ndR}{\sigma\sqrt{\epsilon}})$	$O(d)$
CLVR (**This Paper**)	$O(\frac{\mathrm{nnz}(\boldsymbol{A})R}{\epsilon})$	$O(\frac{\mathrm{nnz}(\boldsymbol{A})R}{\sigma\sqrt{\epsilon}})$	$O(\mathrm{nnz}(\mathrm{row}(\boldsymbol{A})))$

Finally, following the restart strategy based on the *normalized duality gap* for LP introduced in [9], we propose a more straightforward strategy to restart our CLVR algorithm (as well as other iterative algorithms for (PD-GLP)): Restart the algorithm every time a widely known metric for LP optimality [7] halves. Compared with the normalized duality gap, the LP metric can be computed more efficiently and in a more straightforward fashion.

DRO Reformulations. When the loss function is convex, DRO problems with ambiguity sets based on f-divergence [42] or Wasserstein metric [21] are convex. However, as both problems either have complicated constraints or are infinite-dimensional, vanilla first-order methods are inapplicable.

For DRO with f-divergence, we show that by using convex conjugates and introducing auxiliary variables, the problem can be reformulated as a (GLP). As a result, we do not have the issue of biased stochastic gradients encountered in [35] and CLVR can be applied. Even though the resulting problem has larger dimensions, due to the sparseness of the introduced auxiliary variables and the lazy update strategy of CLVR, it can be solved with complexity scaling only with the number of nonzero elements of the data matrix. Due to being cast as a (GLP), the DRO problem can be solved with $O(1/\epsilon)$ iteration complexity with CLVR, while existing methods such as [35] have $O(1/\epsilon^2)$ iteration complexity, with higher iteration cost because of the batch of samples needed to reduce bias. This improvement is enabled in part by considering the primal-dual gap (rather than the primal gap considered in [35]) and by allowing the constraints to be approximately satisfied (see Corollary 1).

For DRO with Wasserstein metric, following the reformulation of [50, Theorem 1], we show further that the problem can be cast in the form of (GLP). Compared with the existing reformulations [21, 36, 50, 58], our reformulation can handle both smooth and nonsmooth convex loss functions. In fact, our reformulation can provide a more compact form for nonsmooth piecewise-linear convex loss functions (such as hinge loss). Moreover, compared with algorithms customized to this problem [36] and extragradient methods [32, 43, 58] for general convex-concave min-max problems, our CLVR method attains the best-known iteration complexity and per-iteration cost, as shown in Table 1.

2 Notation and Preliminaries

For any positive integer p, we use $[p]$ to denote $\{1,2,\ldots,p\}$. We assume that there is a given partition of the set $[n]$ into sets S^j, $j\in[m]$, where $|S^j|=n^j>0$ and $\sum_{j=1}^m n^j=n$. For $j\in[m]$, we use $\boldsymbol{A}^{S^j}$ to denote the submatrix of $\boldsymbol{A}$ with rows indexed by S^j and $\boldsymbol{y}^{S^j}$ to denote the subvector of $\boldsymbol{y}$ indexed by S^j. We use $\mathbf{0}_d$ and $\mathbf{1}_d$ to denote the vectors with all ones and all zeros in d dimensions, respectively. Unless otherwise specified, we use $\|\cdot\|$ to denote the Euclidean norm for vectors and the spectral norm for matrices. For a given proper convex lower semi-continuous function $f:\mathbb{R}\to\mathbb{R}\cup\{+\infty\}$, we define the convex conjugate in the standard way as $f^*(y)=\sup_{x\in\mathbb{R}}\{yx-f(x)\}$ (so that $f^{**}=f$). For a vector $\boldsymbol{u}$, the inequality $\boldsymbol{u}\geq\mathbf{0}$ is applied entry-wise. For a convex function $r(\boldsymbol{x})$, we use $r'(\boldsymbol{x})$ to denote an element of the subdifferential set $\partial r(\boldsymbol{x})$. The proximal operator of $r(\boldsymbol{x})$ over $\mathcal{X}$ is

$$\mathrm{prox}_r(\hat{\boldsymbol{x}})=\arg\min_{\boldsymbol{x}\in\mathcal{X}}\Big\{\frac{1}{2}\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2+r(\boldsymbol{x})\Big\}. \tag{1}$$

Further, we make the following assumptions, which apply throughout the convergence analysis.

Assumption 2. (PD-GLP) *attains at least one primal-dual solution* $(\boldsymbol{x}^*, \boldsymbol{y}^*)$. $\mathcal{W}^*$ *denotes the set of all primal-dual solutions.*

Due to the convex-concave property of (PD-GLP), $\mathcal{W}^*$ is a convex set in $\mathcal{X} \times \mathbb{R}^n$.

Assumption 3. $\hat{L} = \max_{j\in[m]} \|\boldsymbol{A}^{S^j}\|$ *is given at the input, where* $\|\boldsymbol{A}^{S^j}\| = \max_{\|\boldsymbol{x}\|\le 1} \|\boldsymbol{A}^{S^j}\boldsymbol{x}\|$.

Note that $\hat{L}$ can be obtained either via preprocessing of the data or by parameter tuning. By combining Assumptions 1 and 3, it follows that $R \le \hat{L} \le \sqrt{\max_{j\in[m]} |S^j|}R$.

Assumption 4. $r(\boldsymbol{x})$ *is* σ*-strongly convex* $(\sigma \ge 0)$*; that is, for all* $\boldsymbol{x}_1$ *and* $\boldsymbol{x}_2$ *in* $\mathcal{X}$ *and all* $r'(\boldsymbol{x}_2) \in \partial r(\boldsymbol{x}_2)$*, we have* $r(\boldsymbol{x}_1) \ge r(\boldsymbol{x}_2) + \langle r'(\boldsymbol{x}_2), \boldsymbol{x}_1 - \boldsymbol{x}_2\rangle + \frac{\sigma}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2$.

For convex-concave min-max problems, a common metric for measuring solution quality is the primal-dual gap, which, for a feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ of (PD-GLP), is defined by

$$\sup_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{X}\times\mathbb{R}^n} \{\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \boldsymbol{y})\}. \tag{2}$$

However, as the domain of $\boldsymbol{v}$ is unbounded, the primal-dual gap can be infinite, which makes it a poor metric for measuring the progress of algorithms. As a result, for measuring the progress of our algorithm, we consider the following restricted primal-dual gap instead:

$$\sup_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{W}} \{\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \boldsymbol{y})\}, \tag{3}$$

where $\mathcal{W} \subset \mathcal{X} \times \mathbb{R}^n$ is a compact (i.e., closed and bounded) convex set. Such a restricted version of primal-dual gap is standard in the existing literature; see, e.g., [13, 44].

3 The CLVR Algorithm

3.1 Algorithm and Analysis for General Formulation

Algorithm 1 specifies CLVR for (PD-GLP) in the general setting. The algorithm alternates the full update for $\boldsymbol{x}_k$ in Step 4 ($O(d)$ cost) with an incremental block coordinate update for $\boldsymbol{y}_k$ in Steps 5 and 6 (with $O(|S^{j_k}|d)$ cost for dense $\boldsymbol{A}$). The cost of updating auxiliary vectors $\boldsymbol{z}_k$ and $\boldsymbol{q}_k$ is $O(|S^{j_k}|d)$ and $O(d)$, respectively. In essence, CLVR is a primal-dual coordinate method that uses a *dual averaging* update for $\boldsymbol{x}_k$, then updates the state variables $\{\boldsymbol{q}_k\}$ by a *linear recursion*, and computes $\boldsymbol{x}_k$ from $\boldsymbol{q}_{k-1}$ via a *proximal step* without direct dependence on $\boldsymbol{x}_{k-1}$. The output $\tilde{\boldsymbol{x}}_K$ is a convex combination of the iterates $\{\boldsymbol{x}_k\}_{k=1}^K$, as is standard for primal-dual methods. However, $\tilde{\boldsymbol{y}}_K$ is only an *affine* (not convex) combination of $\{\boldsymbol{y}_k\}_{k=0}^K$, as it involves the term $-(m-1)\boldsymbol{y}_0$ (whose coefficient is negative) and some of the coefficients $ma_k - (m-1)a_{k+1}$ multiplying $\boldsymbol{y}_k$ for $k \in \{1, \dots K-1\}$ may also be negative. An affine combination still provides valid bounds because the dual variable vector $\boldsymbol{y}$ appears linearly in (PD-GLP). Moreover, in Step 9, the term $ma_k(\boldsymbol{z}_k - \boldsymbol{z}_{k-1})$ serves to cancel certain errors from the randomization of the update w.r.t. $\boldsymbol{y}_k$, thus playing a key role in implicit variance reduction.

Theorem 1 provides the convergence results for Algorithm 1. The proof is provided in Appendix B. In the theorem (as in the algorithm), γ is a positive parameter that can be tuned.

Theorem 1. *Let* $\boldsymbol{x}_k$*,* $\boldsymbol{y}_k$*,* $k \in [K]$*, be the iterates of Algorithm 1 and let* $\tilde{\boldsymbol{x}}_k$*,* $\tilde{\boldsymbol{y}}_k$ *be defined by*

$$\tilde{\boldsymbol{x}}_k = \frac{1}{A_k}\sum_{i=1}^k a_i\boldsymbol{x}_i, \quad \tilde{\boldsymbol{y}}_k = \frac{1}{A_k}\sum_{i=1}^k (a_i\boldsymbol{y}_i + (m-1)a_i(\boldsymbol{y}_i - \boldsymbol{y}_{i-1})), \tag{4}$$

for $k \in [K]$*. Let* $\mathcal{W}_k \subset \mathcal{X} \times \mathbb{R}^n$*,* $k \in [K]$*, be a sequence of compact convex sets such that* $(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{y}}_k) \in \mathcal{W}_k \subset \mathcal{W} \subset \mathcal{X} \times \mathbb{R}^n$*, where* $\mathcal{W}$ *is also convex and compact. Then:*

$$\begin{aligned}&\mathbb{E}\Big[\sup_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{W}_k} \{\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)\}\Big] \\ &\le \frac{1}{A_k}\bigg(\mathbb{E}\Big[\frac{\gamma}{2}\|\hat{\boldsymbol{u}} - \boldsymbol{x}_0\|^2 + \frac{1}{\gamma}\|\hat{\boldsymbol{v}} - \boldsymbol{y}_0\|^2\Big] + \frac{\gamma}{2}\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2\bigg),\end{aligned} \tag{5}$$

Algorithm 1 Coordinate Linear Variance Reduction (CLVR)

Input: $\boldsymbol{x}_0 \in \mathcal{X}, \boldsymbol{y}_0 \in \mathbb{R}^n, \boldsymbol{z}_0 = \boldsymbol{A}^T\boldsymbol{y}_0, \gamma > 0, \hat{L} > 0, \sigma \geq 0, K, m, \{S^1, S^2, \ldots, S^m\}$.
$a_1 = A_1 = \frac{1}{2\hat{L}m}, \boldsymbol{q}_0 = a_1(\boldsymbol{z}_0 + \boldsymbol{c})$.
for $k = 1, 2, \ldots, K$ **do**
  $\boldsymbol{x}_k = \mathrm{prox}_{\frac{1}{\gamma}A_k r}(\boldsymbol{x}_0 - \frac{1}{\gamma}\boldsymbol{q}_{k-1})$.
  Pick j_k uniformly at random in $[m]$.
  $\boldsymbol{y}_k^{S^i} = \begin{cases} \boldsymbol{y}_{k-1}^{S^i}, & i \neq j_k \\ \boldsymbol{y}_{k-1}^{S^i} + \gamma m a_k\big(\boldsymbol{A}^{S^i}\boldsymbol{x}_k - \boldsymbol{b}^{S^i}\big), & i = j_k \end{cases}$.
  $a_{k+1} = \frac{\sqrt{1+\sigma A_k/\gamma}}{2\hat{L}m}, A_{k+1} = A_k + a_{k+1}$.
  $\boldsymbol{z}_k = \boldsymbol{z}_{k-1} + \boldsymbol{A}^{S^{j_k},T}(\boldsymbol{y}_k^{S^{j_k}} - \boldsymbol{y}_{k-1}^{S^{j_k}})$.
  $\boldsymbol{q}_k = \boldsymbol{q}_{k-1} + a_{k+1}\big(\boldsymbol{z}_k + \boldsymbol{c}\big) + m a_k(\boldsymbol{z}_k - \boldsymbol{z}_{k-1})$.
end for
return $\tilde{\boldsymbol{x}}_K = \frac{1}{A_K}\sum_{k=1}^K a_k\boldsymbol{x}_k, \tilde{\boldsymbol{y}}_K = \frac{1}{A_K}\sum_{k=1}^K (a_k\boldsymbol{y}_k + (m-1)a_k(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}))$.

where $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}) = \arg\sup_{(\boldsymbol{u},\boldsymbol{v})\in\mathcal{W}_k}\{\mathcal{L}(\tilde{\boldsymbol{x}}_k, \boldsymbol{v}) - \mathcal{L}(\boldsymbol{u}, \tilde{\boldsymbol{y}}_k)\}$. *Furthermore,*

$$\mathbb{E}\Big[\frac{\gamma + \sigma A_k}{4}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}_k - \boldsymbol{y}^*\|^2\Big] \leq \frac{\gamma}{2}\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2. \tag{6}$$

Define $K_0 = \lceil \frac{\sigma}{18\hat{L}m\gamma} \rceil$. *Then in the bounds above:*

$$A_k \geq \max\left\{\frac{k}{2\hat{L}m}, \frac{\sigma}{(6\hat{L}m)^2\gamma}\Big(k - K_0 + \max\Big\{3\sqrt{2\hat{L}m\gamma/\sigma}, 1\Big\}\Big)^2\right\}.$$

Observe that $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ in the theorem statement exists because of compactness of $\mathcal{W}_k$ and our assumptions on $r(\cdot)$. The parameter γ can be tuned to balance the relative weights of primal and dual initial quantities $\|\boldsymbol{x}^* - \boldsymbol{x}_0\|$ and $\|\boldsymbol{y}^* - \boldsymbol{y}_0\|$ (or estimates of these quantities), which can significantly influence practical performance of the method.

In addition to the guarantee on the variational form, due to the linear structure, we also provide explicit guarantees for both the objective and the constraints in (GLP), stated in the following corollary.

Corollary 1. *In Algorithm 1, for all* $k \geq 1$, $\tilde{\boldsymbol{x}}_k$ *satisfies*

$$\mathbb{E}[\|\boldsymbol{y}^*\| \cdot \|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\|] \leq \frac{\gamma\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2 + \frac{1}{\gamma}\mathbb{E}[\|\boldsymbol{v} - \boldsymbol{y}_0\|^2]}{A_k},$$

$$|\mathbb{E}[(\boldsymbol{c}^T\tilde{\boldsymbol{x}}_k + r(\tilde{\boldsymbol{x}}_k)) - (\boldsymbol{c}^T\boldsymbol{x}^* + r(\boldsymbol{x}^*))]| \leq \frac{\gamma\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 + \frac{1}{2\gamma}\|\boldsymbol{y}^* - \boldsymbol{y}_0\|^2 + \frac{1}{\gamma}\mathbb{E}[\|\boldsymbol{v} - \boldsymbol{y}_0\|^2]}{A_k},$$

where $\boldsymbol{v} = 2\frac{\|\boldsymbol{y}^*\|}{\|\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b}\|}(\boldsymbol{A}\tilde{\boldsymbol{x}}_k - \boldsymbol{b})$.

In CLVR, we allow for arbitrary $(\boldsymbol{x}_0, \boldsymbol{y}_0) \in \mathcal{X} \times \mathbb{R}^n$. Nevertheless, by setting $\boldsymbol{y} = \boldsymbol{0}_n$, we can obtain $\boldsymbol{z}_0 = \boldsymbol{0}_d$ at no cost — a useful strategy for large-scale problems since it avoids the (potentially expensive) single matrix-vector multiplication w.r.t. $\boldsymbol{A}$.

3.2 Lazy Update for Sparse and Structured (PD-GLP)

In Algorithm 1, direct computation of the iterates $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ and the output points $(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{y}}_k)$ can be expensive. However, [16] showed that it is possible to only update the averaged vector in the coordinate block chosen for that iteration. This strategy requires us to record the most recent update for each coordinate block and update it only when it is selected again, which is tricky and needs to be implemented carefully. For sparse and block coordinate-separable instances of (PD-GLP), we show that by introducing auxiliary variables that are sparsely connected, we can significantly simplify CLVR and make its complexity scale independently of the ambient dimension $n \cdot d$, instead scaling with $\mathrm{nnz}(\boldsymbol{A})$. Due to space constraints, we defer technical details, including the lazy version of CLVR and associated proofs, to Appendix A.

3.3 Restart Scheme

We now propose a fixed restart strategy with a fixed number of iterations per each restart epoch and discuss an adaptive restart strategy for the special case of standard-form LP, which corresponds to (GLP) with $r(\boldsymbol{x}) \equiv 0$ and $\mathcal{X} = \{\boldsymbol{x} : x_i \geq 0, i \in [d]\}$. We write

$$\min_{\boldsymbol{x}} \ \boldsymbol{c}^T \boldsymbol{x} \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \ \boldsymbol{x} \geq \boldsymbol{0}_d, \tag{LP}$$

and the primal-dual form

$$\min_{\boldsymbol{x} \geq \boldsymbol{0}_d} \max_{\boldsymbol{y} \in \mathbb{R}^n} \Big\{ \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{c}^T \boldsymbol{x} + \boldsymbol{y}^T \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}^T \boldsymbol{b} \Big\}. \tag{PD-LP}$$

This problem has a sharpness property that can be used to obtain linear convergence in first-order methods [9]. For convenience, in the following, we define $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{y}), \hat{\boldsymbol{w}} = (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}), \tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ and $\boldsymbol{w}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$. Meanwhile, for $\gamma > 0$, we denote the weighted norm $\|\boldsymbol{w}\|_{(\gamma)} := \sqrt{\gamma \|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 + \frac{1}{\gamma}\|\boldsymbol{y} - \boldsymbol{y}^*\|_2^2}$. Further, we use $\mathcal{W}^*$ to denote the optimal solution set of the LP and define the distance to $\mathcal{W}^*$ by $\text{dist}(\boldsymbol{w}, \mathcal{W}^*)_{(\gamma)} = \min_{\boldsymbol{w}^* \in \mathcal{W}^*} \|\boldsymbol{w} - \boldsymbol{w}^*\|_{(\gamma)}$. When $\gamma = 1$, $\|\cdot\|_{(\gamma)}$ is the standard Euclidean norm. Then based on (PD-LP), we can use the following classical LP metric[3] to measure the progress of iterative algorithms for LP:

$$\begin{aligned} &\text{LPMetric}(\boldsymbol{x}, \boldsymbol{y}) \\ &= \sqrt{\|\max\{-\boldsymbol{x}, \boldsymbol{0}\}\|_2^2 + \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \|\max\{-\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{c}, \boldsymbol{0}\}\|_2^2 + |\max\{\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{y}, 0\}|^2}, \end{aligned} \tag{7}$$

which can be explicitly and directly computed. For the Euclidean case ($\gamma = 1$), it is well-known [28] that there exists a Hoffman constant H_1 such that

$$\text{LPMetric}(\boldsymbol{w}) \geq H_1 \text{dist}(\boldsymbol{w}, \mathcal{W}^*)_{(1)}. \tag{8}$$

Using the equivalence of norms in finite dimensions, for general $\gamma > 0$, we can conclude that there exists another constant H_γ (to which we refer as the generalized Hoffman's constant) such that

$$\text{LPMetric}(\boldsymbol{w}) \geq H_\gamma \text{dist}(\boldsymbol{w}, \mathcal{W}^*)_{(\gamma)}. \tag{9}$$

Using Eq. (9) and Theorem 1, we then obtain the following bounds for distance and LPMetric.

Theorem 2. *Consider the* CLVR *algorithm applied to the standard-form LP problem* (PD-LP)*, with input $\boldsymbol{w}_0$ and output $\tilde{\boldsymbol{w}}_k$. Given $\gamma > 0$, define $\boldsymbol{w}^* = \arg\min_{w \in \mathcal{W}^*} \|\boldsymbol{w}_0 - \boldsymbol{w}\|_{(\gamma)}$, and define $C_0 = \gamma + 1/\gamma + (\sqrt{2} + 1)\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_{(\gamma)} + \|\boldsymbol{w}^*\|_{(\gamma)}$. Then for H_γ defined as in* (9)*, we have*

$$\mathbb{E}\Big[\sqrt{\text{dist}(\tilde{\boldsymbol{w}}_k, \mathcal{W}^*)_{(\gamma)}}\,\Big] \leq 5\sqrt{\frac{\hat{L}mC_0}{H_\gamma k}}\sqrt{\text{dist}(\boldsymbol{w}_0, \mathcal{W}^*)_{(\gamma)}},$$

$$\mathbb{E}\big[\sqrt{\text{LPMetric}(\tilde{\boldsymbol{w}}_k)}\big] \leq 5\sqrt{\frac{\hat{L}mC_0}{H_\gamma k}}\sqrt{\text{LPMetric}(\boldsymbol{w}_0)}.$$

As a result, by Theorem 2, if we know the values of $\hat{L}$, $\|\boldsymbol{w}^*\|_{(\gamma)}$ and H_γ, then by setting $k = \frac{100\hat{L}mC_0}{H_\gamma k}$, we can halve the square root of the distance and the LPMetric in expectation. Thus we can obtain linear convergence if we restart the CLVR algorithm after a fixed number of iterations. However, the values of $\|\boldsymbol{w}^*\|_{(\gamma)}$ and H_γ are often unknown and thus make this strategy unrealistic in practice.

Compared with the above fixed restart strategy, a natural strategy is to restart whenever the LPMetric halves (summarized in Algorithm 4 in the appendix). Since LPMetric is easy to monitor and update, implementation of this strategy is straightforward. However, bounding the number of iterations required to halve the metric (in expectation or with high probability) seems nontrivial. What can be said (based on Theorem 2 and denoting by K the number of iterations on CLVR between restarts) is that $\mathbb{P}[K > \frac{50\hat{L}mC_0}{\delta^2 H_\gamma}] \leq \delta$. This follows by Markov inequality, as $\mathbb{P}[K > k] = \mathbb{P}\Big[\sqrt{\text{LPMetric}(\tilde{\boldsymbol{w}}_k)} > \sqrt{\frac{\text{LPMetric}(\boldsymbol{w}_0)}{2}}\Big] \leq 5\sqrt{2\frac{\hat{L}mC_0}{H_\gamma k}}$. Although we use adaptive restart in our experiments, we defer its convergence analysis to future work. Finally, as an independent and parallel work to ours, [38] proposed a high probability guarantee for adaptive restart for stochastic extragradient-type methods.

[3] In (PD-LP), we dualize the constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by $\boldsymbol{y}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$ instead of $\boldsymbol{y}^T(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})$, so in our LP metric, there exist a sign difference for $\boldsymbol{y}$ from the more common representation such as the one in [9].

4 Application: DRO

Consider sample vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$ with labels $\{b_1, b_2, \ldots, b_n\}$, where $b_i \in \{1, -1\}$ $(i \in [n])$. The DRO problem with f-divergence based ambiguity set is

$$\min_{\boldsymbol{x} \in \mathcal{X}} \sup_{\boldsymbol{p} \in \mathcal{P}_{\rho,n}} \sum_{i=1}^{n} p_i g(b_i \boldsymbol{a}_i^T \boldsymbol{x}), \tag{10}$$

where $\mathcal{P}_{\rho,n} = \left\{\boldsymbol{p} \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1, p_i \geq 0\,(i \in [n]), D_f(\boldsymbol{p} \| \mathbf{1}/n) \leq \frac{\rho}{n}\right\}$ is the ambiguity set, g is a convex loss function and D_f is an f-divergence defined by $D_f(\boldsymbol{p} \| \boldsymbol{q}) = \sum_{i=1} q_i f(p_i/q_i)$ with $\boldsymbol{p}, \boldsymbol{q} \in \left\{\boldsymbol{p} \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1, p_i \geq 0\right\}$ and f being a convex function [42]. The formulation (10) is a nonbilinearly coupled convex-concave min-max problem with constraint set $\mathcal{P}_{\rho,n}$ for which efficient projections are not available in general. When g is a nonsmooth loss (e.g., the hinge loss), many well-known methods such as the extragradient [32, 43] cannot be used even if we could project onto $\mathcal{P}_{\rho,n}$ efficiently. However, by introducing auxiliary variables, additional linear constraints, and simple convex constraints, we can make the interacting term between primal and dual variables bilinear, as shown next. (See Appendix C for a proof.)

Theorem 3. *Let $\mathcal{X}$ be a compact convex set. Then the DRO problem in Eq.* (10) *is equivalent to*

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{q},\gamma} \quad & \left\{\gamma + \frac{\rho\mu_1}{n} + \frac{1}{n}\sum_{i=1}^{n} \mu_i f^*\left(\frac{q_i}{\mu_i}\right)\right\} \\
\text{s.t.} \quad & \boldsymbol{w} + \boldsymbol{v} - \frac{\boldsymbol{q}}{n} - \gamma \mathbf{1}_n = \mathbf{0}_n, \\
& u_i = b_i \boldsymbol{a}_i^T \boldsymbol{x}, && i \in [n] \\
& \mu_1 = \mu_2 = \cdots = \mu_n, \\
& g(u_i) \leq w_i, && i \in [n] \\
& q_i \in \mu_i \operatorname{dom}(f^*), && i \in [n] \\
& v_i \geq 0,\ \mu_i \geq 0, && i \in [n] \\
& \boldsymbol{x} \in \mathcal{X}.
\end{aligned}$$

In Theorem 3, the domain of the one-dimensional convex function $f^*(\cdot)$ is an interval such as $[a, b]$, so that $q_i \in \mu_i \operatorname{dom}(f^*)$ denotes the inequality $\mu_i a \leq q_i \leq \mu_i b$. Since the perspective function $\mu f^*\left(\frac{q}{\mu}\right)$ is a simple convex function of two variables, we can assume that the proximal operator for this function on the domain $\{(\mu, q) : q \in \mu \operatorname{dom}(f^*), \mu > 0\}$ can be computed efficiently [11]. Similarly, we can assume that the constraint $g(u) \leq w$ admits an efficiently computable projection operator. As a result, the formulation (10) can be solved by CLVR. When expressing (10) in the form of (PD-GLP), the primal and dual variable vectors have dimensions $d + 1 + 4n$ and $3n - 1$, respectively. However, according to Table 1, provided that $\mathcal{X}$ is coordinate separable, the overall complexity of CLVR will only be $O\left(\frac{(\operatorname{nnz}(\boldsymbol{A})+n)(R+1)}{\epsilon}\right)$.

The original DRO problem with Wasserstein metric based ambiguity set is an *infinite*-dimensional *nonbilinearly* coupled convex-concave min-max problem defined by

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \sup_{\mathbb{P} \in \mathcal{P}_{\rho,\kappa}} \mathbb{E}^{\mathbb{P}}[g(b\boldsymbol{a}^T \boldsymbol{w})], \tag{11}$$

where $\boldsymbol{a} \in \mathbb{R}^d, b \in \{1, -1\}$, $\mathbb{P}$ is a distribution on $\mathbb{R}^d \times \{1, -1\}$, g is a convex loss function and $\mathcal{P}_{\rho,\kappa}$ is the Wasserstein metric-based ambiguity set [50]. Our reformulation for Eq. (11) is in Appendix C.2.

5 Numerical Experiments and Discussion

We provide experimental evaluations of our algorithm for the reformulation of the DRO with Wasserstein metric of ℓ_1-norm (with $\kappa = 0.1$ and $\rho = 10$) and hinge loss. Full details for the experimental setting can be found in Appendix D.

Comparison with Primal-Dual Algorithms. Figure 1 provides an algorithm comparison in terms of the number of data passes and wall-clock time. The spikes in all the plots are due restarts; at the beginning of each cycle, the value of LP metric increases significantly, then decreases rapidly. For

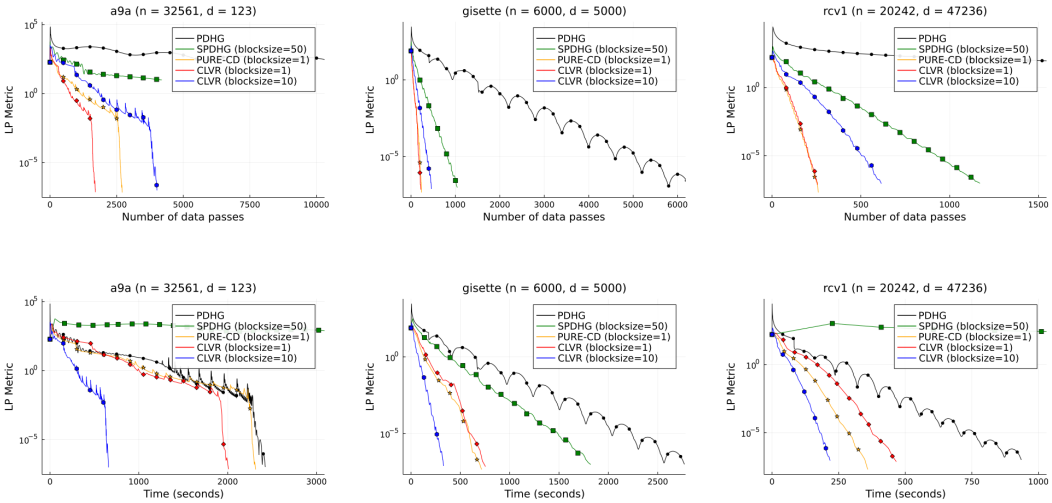

Figure 1: Comparison of numerical results in terms of number of data passes and wall-clock time.

Table 2: Comparison of numerical results against off-the-shelf solvers. Each value represents the time required by a solver/algorithm to reach accuracy $< 10^{-8}$.

Time (seconds)	Reformulated a9a $d = 130738, n = 97929$	Reformulated gisette $d = 44002, n = 28000$	Reformulated rcv1 $d = 269914, n = 155198$
JuMP+GLPK	899	$> 4 \times 10^4$	$> 4 \times 10^4$
JuMP+Gurobi(simplex)	893	2482	7008
JuMP+Gurobi(barrier)	**26**	1039.7	1039.5
CLVR	962	**697**	**582**

the number of data passes (top row), CLVR with block size 1 and PURE-CD perform best on all three datasets. CLVR with block size 10 and SPDHG with block size 50 have second-tier performance, while PDHG is worst. For the CLVR algorithm, smaller block size corresponds to smaller $\hat{L}$ in Assumption 3, which corresponds to better complexity in terms of data passes by Theorem 1. Nevertheless, the gap between empirical performance and theoretical guarantee for SPDHG and PURE-CD deserves further research because, to date, they have only been shown to have the same iteration complexity as PDHG. Empirically on `a9a`, CLVR with block size 1 performs better than PURE-CD in terms of data passes.

In terms of wall-clock time (bottom row), because of different per-iteration costs of each algorithm and instruction-level parallelism in modern processors [26], the plots differ significantly from the plots for number of data passes. Even with block size 50, SPDHG spends the most wall-clock time for one data pass and is the slowest on sparse datasets `a9a` and `rcv1`, but is faster than PDHG on the dense dataset `gisette`. Meanwhile, while CLVR with block size 10 is not best in terms of data passes, it remains fastest in terms of wall-clock time on all datasets due to cheaper per-iteration cost and instruction-level parallelism. On `rcv1`, the per-iteration cost of PURE-CD is about 60% of that of CLVR with block size 1. Hence, despite having similar performance in terms of data passes, PURE-CD is faster than CLVR with block size 1, but is still slower than CLVR with block size 10.

Comparison with Off-the-shelf Solvers. Table 2 shows that the performance of CLVR is competitive against state-of-the-art solvers such as GLPK [1] and Gurobi [24]. Moreover, we observe that CLVR reached accurate solutions significantly faster than GLPK and Gurobi in the reformulated problems with `gisette` and `rcv1` datasets. Although CLVR is much slower than Gurobi(barrier) on `a9a` datasets, we believe that much of the performance gap in this case is due to the poor conditioning of this instance and the use of preconditioners in the barrier method. We leave preconditioning and other heuristic speedups of CLVR for future work.

CLVR is fastest in both the number of data passes and wall-clock time among all primal-dual algorithms we compared against on these datasets and it is also competitive with off-the-shelf solvers. Since it has the best known theoretical guarantee, we believe that CLVR could be a method of choice.

References

[1] GLPK (GNU linear programming kit). `https://www.gnu.org/software/glpk/`, 2022.

[2] A. Alacaoglu and Y. Malitsky. Stochastic variance reduction for variational inequality methods. *arXiv preprint arXiv:2102.08352*, 2021.

[3] A. Alacaoglu, Q. T. Dinh, O. Fercoq, and V. Cevher. Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization. In *Proc. NIPS'17*, 2017.

[4] A. Alacaoglu, O. Fercoq, and V. Cevher. On the convergence of stochastic primal-dual hybrid gradient. *arXiv preprint arXiv:1911.00799*, 2019.

[5] A. Alacaoglu, O. Fercoq, and V. Cevher. Random extrapolation for primal-dual coordinate descent. In *Proc. ICML'20*, 2020.

[6] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proc. ACM STOC'17*, 2017.

[7] E. D. Andersen and K. D. Andersen. The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High performance optimization*, pages 197–232. Springer, 2000.

[8] D. Applegate, M. Díaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. *arXiv preprint arXiv:2106.04756*, 2021.

[9] D. Applegate, O. Hinder, H. Lu, and M. Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *arXiv preprint arXiv:2105.12715*, 2021.

[10] P. Bianchi. Ergodic convergence of a stochastic proximal point algorithm. *SIAM Journal on Optimization*, 26(4):2235–2260, 2016.

[11] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[12] Y. Carmon, Y. Jin, A. Sidford, and K. Tian. Variance reduction for matrix games. In *Proc. NeurIPS'19*, 2019.

[13] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[14] A. Chambolle, M. J. Ehrhardt, P. Richtárik, and C.-B. Schonlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018.

[15] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[16] C. D. Dang and G. Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization*, 25(2):856–881, 2015.

[17] D. P. De Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003.

[18] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956.

[19] J. Duchi and H. Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

[20] J. C. Duchi, P. W. Glynn, and H. Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

[21] P. M. Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[22] O. Fercoq and P. Bianchi. A coordinate-descent primal-dual algorithm with large step size and possibly nonseparable functions. *SIAM Journal on Optimization*, 29(1):100–134, 2019.

[23] G. O. Gurobi. Reference manual, gurobi optimization, 2020.

[24] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022. URL `https://www.gurobi.com`.

[25] E. Y. Hamedani and A. Jalilzadeh. A stochastic variance-reduced accelerated primal-dual method for finite-sum saddle-point problems. *arXiv preprint arXiv:2012.13456*, 2020.

[26] J. L. Hennessy and D. A. Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011.

[27] N. Ho-Nguyen and S. J. Wright. Adversarial classification via distributional robustness with Wasserstein ambiguity. *arXiv preprint arXiv:2005.13815*, November 2021.

[28] A. J. Hoffman. On approximate solutions of systems of linear inequalities. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 174–176. World Scientific, 2003.

[29] W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *Proc. ICML'18*, 2018.

[30] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. NIPS'13*, 2013.

[31] A. Juditsky, A. Nemirovski, and C. Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[32] G. Korpelevich. The extragradient method for finding saddle points and other problems. *Ekonomika i Matematicheskie Metody*, 12(5):747–756 (in Russian; English translation in Matekon), 1976.

[33] P. Latafat, N. M. Freris, and P. Patrinos. A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization. *IEEE Transactions on Automatic Control*, 64 (10):4050–4065, 2019.

[34] S. Lee and S. J. Wright. Manifold identification in dual averaging for regularized stochastic online learning. *Journal of Machine Learning Research*, 13(6), 2012.

[35] D. Levy, Y. Carmon, J. C. Duchi, and A. Sidford. Large-scale methods for distributionally robust optimization. In *Proc. NeurIPS'20*, 2020.

[36] J. Li, S. Huang, and A. M.-C. So. A first-order algorithmic framework for wasserstein distributionally robust logistic regression. *arXiv preprint arXiv:1910.12778*, 2019.

[37] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer, et al. Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4, 2020.

[38] H. Lu and J. Yang. Nearly optimal linear convergence of stochastic primal-dual methods for linear programming. *arXiv preprint arXiv:2111.05530*, 2021.

[39] O. Mangasarian. A newton method for linear programming. *Journal of Optimization Theory and Applications*, 121(1):1–18, 2004.

[40] O. L. Mangasarian. Normal solutions of linear programs. In *Mathematical Programming at Oberwolfach II*, pages 206–216. Springer, 1984.

[41] O. L. Mangasarian and R. Meyer. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, 1979.

[42] H. Namkoong and J. C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Proc. NIPS'16*, 2016.

[43] A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[44] Y. Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2-3):319–344, 2007.

[45] H. Ouyang, N. He, L. Tran, and A. Gray. Stochastic alternating direction method of multipliers. In *Proc. ICML'13*, 2013.

[46] Y. Ouyang and Y. Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, pages 1–35, 2019.

[47] A. Patrascu and I. Necoara. Nonasymptotic convergence of stochastic proximal point methods for constrained convex optimization. *Journal of Machine Learning Research*, 18(1):7204–7245, 2017.

[48] H. Rahimian and S. Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[49] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[50] S. Shafieezadeh Abadeh, P. Mohajerin Esfahani, and D. Kuhn. Distributionally robust logistic regression. In *Proc. NIPS'15*, 2015.

[51] A. Shapiro. On duality theory of conic linear problems. In *Semi-infinite programming*, pages 135–165. Springer, 2001.

[52] C. Song, Y. Jiang, and Y. Ma. Variance reduction via accelerated dual averaging for finite-sum optimization. In *Proc. NeurIPS'20*, 2020.

[53] C. Song, S. J. Wright, and J. Diakonikolas. Variance reduction via primal-dual accelerated dual averaging for nonsmooth convex finite-sums. In *Proc. ICML'21*, 2021.

[54] J. Stoer. Duality in nonlinear programming and the minimax theorem. *Numerische Mathematik*, 5(1):371–379, 1963.

[55] C. Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[56] W. Wiesemann, D. Kuhn, and M. Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

[57] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

[58] Y. Yu, T. Lin, E. Mazumdar, and M. I. Jordan. Fast distributionally robust learning with variance reduced min-max optimization. *arXiv preprint arXiv:2104.13326*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] All theoretical claims are supported by proofs, which, due to space constraints, are provided in the appendix. Numerical results are provided in Section 5, with full technical details of the experiments provided in Appendix D.

(b) Did you describe the limitations of your work? [Yes] The three main examples are: (i) the discussion of lazy update strategy and guarantees for $\hat{\boldsymbol{x}}_k$ versus $\tilde{\boldsymbol{x}}_k$ in Appendix A; (ii) the discussion of linear convergence under the restart strategy and theoretical guarantees for scheduled restart versus empirical guarantees for adaptive restart in Section 3.3; and (iii) comparison to off-the-shelf solvers in Section 5 and use of preconditioning.

(c) Did you discuss any potential negative societal impacts of your work? [N/A] This work concerns a generic solver for generalized linear programs and as such is not tied to any specific application that could have a potential negative societal impact.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] See assumptions in Section 2.

(b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B and C for complete proofs.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include our code in the supplementary material.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Section 5 and Appendix D.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Error bars are not reported due to almost identical behaviors and results from the large number of stochastic iterations.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In Appendix D.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We used datasets from LibSVM, which is a open source machine learning library and cited the creators [15] in Appendix D which provides full experimental details.

(b) Did you mention the license of the assets? [N/A] LibSVM datasets are distributed under the BSD license which is "a low restriction type of license for open source software that does not put requirements on redistribution."

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Our code is provided in the supplementary material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] As mentioned before, the datasets we use are open source.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]