Value Function Based Difference-of-Convex Algorithm for Bilevel Hyperparameter Selection Problems

Lucy Gao [1] **Jane J. Ye** [2] **Haian Yin** [3] **Shangzhi Zeng** [2] **Jin Zhang** [3 4]

Abstract

Gradient-based optimization methods for hyperparameter tuning guarantee theoretical convergence to stationary solutions when for fixed upper-level variable values, the lower level of the bilevel program is strongly convex (LLSC) and smooth (LLS). This condition is not satisfied for bilevel programs arising from tuning hyperparameters in many machine learning algorithms. In this work, we develop a sequentially convergent Value Function based Difference-of-Convex Algorithm with inexactness (VF-iDCA). We show that this algorithm achieves stationary solutions without LLSC and LLS assumptions for bilevel programs from a broad class of hyperparameter tuning applications. Our extensive experiments confirm our theoretical findings and show that the proposed VF-iDCA yields superior performance when applied to tune hyperparameters.

1. Introduction

Virtually all machine learning algorithms require hyperparameter tuning. For example, in regression and classification problems, regularization is often used to induce structured solutions and to control model complexity. The degree of regularization is typically controlled by hyperparameters. Careful tuning of these hyperparameters is critical to the predictive accuracy of the fitted model.

An effective hyperparameter selection strategy is to choose the hyperparameters that minimize a loss function on a held-out validation set, which amounts to solving a bilevel program (BLP). We consider the following BLP framework:

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\lambda\in\mathbb{R}^J_+} \quad & L(x) \\ \text{s.t.} \quad & x \in \arg\min_{x'\in\mathbb{R}^n} \left\{ l(x') + \sum_{i=1}^{J} \lambda_i P_i(x') \right\}, \end{aligned} \tag{1}$$

where $L, l : \mathbb{R}^n \to \mathbb{R}$ and $P_i : \mathbb{R}^n \to \mathbb{R}_+,\ i = 1, \ldots, J$ are merely convex (possibly non-smooth) functions. In BLP Eq. (1), λ is a vector of hyperparameters, the upper-level (UL) problem minimizes the validation error in terms of the hyperparameters, and the lower-level (LL) problem fits a model to training data for a given choice of hyperparameters. Note that in BLP Eq. (1), we assume that for each $\lambda \in \mathbb{R}^J_+$, the LL problem has solutions.

Table 1 presents examples of bilevel hyperparameter selection problems of the form (1) with non-strongly convex and/or non-smooth LL objective functions. In Section 4, we will study a generalization of BLP Eq. (1) with LL constraints, which captures the support vector machine classification example in Table 1.

1.1. Related Work

The simplest hyperparameter selection approach is brute-force grid search, which is seldom applied when there are more than two hyperparameters due to its unfavourable computational complexity. The current gold standard in the presence of more than two hyperparameters is Bayesian optimization, which models the conditional probability of the performance on some metric given hyperparameters and a dataset. However, these gradient-free methods often scale poorly beyond 10 to 20 hyperparameters.

Gradient-based optimization methods (GM) can handle up to a few hundred hyperparameters. Existing gradient-based methods can be roughly divided into two categories based on the strategy used for calculating the hypergradient (the gradient of the UL objective). Implicit gradient methods (IGM), also known as implicit differentiation (Pedregosa, 2016; Rajeswaran et al., 2019; Lorraine et al., 2020), replace the LL sub-problem with an implicit equation. These IGM apply the implicit function theorem (IFT) to the optimal-

[1] Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, Canada [2] Department of Mathematics and Statistics, University of Victoria, Victoria, British Columbia, Canada [3] Department of Mathematics, SUSTech International Center for Mathematics, Southern University of Science and Technology, Shenzhen, Guangdong, China [4] National Center for Applied Mathematics Shenzhen, Shenzhen, Guangdong, China. Correspondence to: Jin Zhang <zhangj9@sustech.edu.cn>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

Table 1. Examples of bilevel hyperparameter selection problems of the form (1) and its generalization. These examples were also explored in Kunapuli et al. (2008) and Feng & Simon (2018).

Machine learning algorithm	x	λ	$L(x)/l(x)$	$\sum_{i=1}^{J}\lambda_i P_i(x)$	LL Constraints
elastic net	$\boldsymbol{\beta}$	λ_1, λ_2	$\frac{1}{2}\sum_{i\in I_{\text{val}}/i\in I_{\text{tr}}} \lvert b_i - \boldsymbol{\beta}^\top \mathbf{a}_i\rvert^2$	$\lambda_1\lVert\boldsymbol{\beta}\rVert_1 + \frac{\lambda_2}{2}\lVert\boldsymbol{\beta}\rVert_2^2$	-
sparse group lasso	$\boldsymbol{\beta}$	$\lambda \in \mathbb{R}_+^{M+1}$	$\frac{1}{2}\sum_{i\in I_{\text{val}}/i\in I_{\text{tr}}} \lvert b_i - \boldsymbol{\beta}^\top \mathbf{a}_i\rvert^2$	$\sum_{m=1}^{M}\lambda_m\lVert\boldsymbol{\beta}^{(m)}\rVert_2 + \lambda_{M+1}\lVert\boldsymbol{\beta}\rVert_1$	-
low-rank matrix completion	$\boldsymbol{\theta}$, $\boldsymbol{\beta}$, Γ	$\lambda \in \mathbb{R}_+^{2G+1}$	$\sum_{(i,j)\in\Omega_{\text{val}}/(i,j)\in\Omega_{\text{tr}}} \frac{1}{2}\lvert M_{ij} - \mathbf{x}_i\boldsymbol{\theta} - \mathbf{z}_j\boldsymbol{\beta} - \Gamma_{ij}\rvert^2$	$\lambda_0\lVert\Gamma\rVert_* + \sum_{g=1}^{G}\lambda_g\lVert\boldsymbol{\theta}^{(g)}\rVert_2 + \sum_{g=1}^{G}\lambda_{g+G}\lVert\boldsymbol{\beta}^{(g)}\rVert_2$	-
support vector machine	$\mathbf{w}$, c	λ, $\bar{\mathbf{w}}$	$\sum_{j\in I_{\text{val}}/j\in I_{\text{tr}}} \max(1 - b_j(\mathbf{w}^\top\mathbf{a}_j - c), 0)$	$\frac{\lambda}{2}\lVert\mathbf{w}\rVert^2$	$-\bar{\mathbf{w}} \le \mathbf{w} \le \bar{\mathbf{w}}$

The detailed descriptions of these problems are presented in the Supplemental Material.

ity conditions of the LL problem, and hence derive their hypergradients by solving a linear system. As it involves computing a Hessian matrix and its inverse, in practice, the conjugate gradient method (CG) or Neumann method are used for fast inverse computation. Explicit gradient methods (EGM) replace the LL problem with a gradient-descent-type dynamics iteration (Franceschi et al., 2017; 2018; Finn et al., 2017; Nichol et al., 2018; Lorraine & Duvenaud, 2018). Specifically, Franceschi et al. (2017) and Franceschi et al. (2018) first calculate gradient representations of the LL objective and then perform either reverse or forward gradient computations (called Reverse Hyper-Gradient (RHG) and Forward Hyper-Gradient (FHG)) for the UL objective. In order to reduce the amount of computation, Shaban et al. (2019) proposes truncated reverse hypergradient to truncate the gradient trajectory, and Liu et al. (2019a) uses the difference of vectors to approximate the gradient. Recently, Ji et al. (2020a) and Ji et al. (2021) analyze the convergence rate and complexity of IGM and EGM, respectively. Ji et al. (2022) analyze the effect of inner loop step for IGM and EGM. Besides, some work (Liu et al., 2020; 2021a; Sow et al., 2022; Liu et al., 2022) study the case where LL problem is convex but not strongly convex; Liu et al. (2021b) proposed an initialization auxiliary method for the setting where the LL problem is generally nonconvex.

1.2. Our Motivations and Contributions

Some theoretical progress has been made in both gradient-free and gradient-based methods. However, existing GM require restrictive assumptions to ensure the convergence towards stationarity: both IGM and EGM require the LL strong convexity (LLSC) and smoothness (LLS), which is often violated in applications. For IGM, the subsequential convergence of stationary solutions can be found in Pedregosa (2016). Without the LLS assumption, some IGM variants study the computation of hypergradients, where the convergence analysis of stationarity of such methods is still lacking; see Implicit Gradient-based Joint Optimization (IGJO) in Feng & Simon (2018) and Implicit Forward Differentiation Method (IFDM) in Bertrand et al. (2020) and Bertrand et al. (2021). For EGM, when the LL objective is strongly convex uniformly for λ and its gradient w.r.t. x is uniformly Lipschitz continuous, the LL gradient iterates converge linearly in a uniform rate. This further leads to a uniformly linear convergence of the hypergradient of approximated UL objective to the true hypergradient. The subsequential convergence towards stationary solutions for EGM thus follows easily; see (Grazzi et al., 2020). Focusing on stationarity convergence, under LLSC and LLS, Ji et al. (2020b) and Ji & Liang (2021) give a comprehensive characterization on the nonasymptotic convergence rate and complexity bounds for both IGM and EGM.

Many applications involve BLP where *LLSC and LLS are violated*; see the hyperparameter selection applications in BLP Eq. (1) and Table 1. This motivates us to investigate algorithms with provable convergence towards stationary solutions *without* assuming LLSC and LLS.

Table 2. Comparing theoretical results among BLP algorithms.

Category	Methods	LLSC	LLS	Conv. Type	Sol. Quality
EGM	FHG/RHG	w/	w/	subsequential	stationary
IGM	CG/Neumann	w/	w/	subsequential	stationary
	IGJO/IFDM	w/	w/o	-	-
DCA	VF-iDCA	w/o	w/o	sequential	stationary

A striking feature of our study is the decoupling of the hyperparameters from the regularization term, which results in an equivalent BLP reformulation with fully convex LL problem. Critically, since the reformulated LL problem is fully convex, the value function (VF) of the LL problem is convex. Subsequently, we can further reformulate BLP Eq. (1) as a DC program. By using the main idea of DC algorithm (DCA) which linearizes the concave part of the DC structure, we propose a sequential convex programming scheme with subproblem inexactness, named VF-iDCA. For a wide range of applications, we justify the sequential convergence towards stationary solutions, which is an unusual finding in BLP optimization. Recently, a DC-type algorithm, namely, iP-DCA for solving bilevel support vector classification problem was proposed in Ye et al. (2021). In comparison with iP-DCA, the new hyperparameter decoupling mechanism of VF-iDCA allows us to efficiently handle more hyperparameter selection BLPs where the LL problems are equipped with complex regularization terms. Our main contributions are as follows:

- By decoupling hyperparameters from the regularization, based on the value function approach, VF-iDCA constructs a series of tractable convex subproblems.

- To our best knowledge, we establish the first strict sequential convergence towards stationary solutions for an important class of hyperparameter selection applications without assuming LLSC and LLS for LL tasks.
- We conduct experiments to verify our theoretical findings and evaluate VF-iDCA on various hyperparameter selection BLPs. Our results compare favourably to existing gradient-free and gradient-based methods.

In Table 2, we summarize our contributions to the theory of convergence of BLP algorithms by comparing our results to some existing results in the literature.

2. Value Function Based Difference-of-Convex Algorithm with Inexactness

2.1. Hyperparameter Decoupling and Fully Convex LL

The LL problem in Eq. (1) is a penalized problem with regularization term involving coupled UL and LL variables:

$$\min_{x'} \; l(x') + \sum_{i=1}^{J} \lambda_i P_i(x'). \tag{2}$$

We decouple the hyperparameter variables λ from the regularization term by introducing a new variable r, which yields the following constrained optimization problem:

$$\min_{x'} \; l(x') \quad \text{s.t. } P_i(x') \le r_i, \; i = 1, \dots, J. \tag{3}$$

Denote their solution sets by $S_p(\lambda)$ and $S_c(r)$, respectively. Thanks to the LL convexity w.r.t. x, there is a one-to-one relationship between regularized problem Eq. (2) and constrained problem Eq. (3). That is, for any $\lambda \ge 0$, there is $r \ge 0$ such that $S_p(\lambda) = S_c(r)$ and vice versa. This suggests working with the following BLP:

$$\begin{aligned} \min_{x, r \in \mathbb{R}^J_+} \quad & L(x) \\ \text{s.t.} \quad & x \in \arg\min_{x'} \{ l(x') \text{ s.t. } P_i(x') \le r_i, i = 1, \dots, J \}. \end{aligned} \tag{4}$$

In the section 2.2, we will use the fact that the LL of Eq. (4) is *fully convex w.r.t the joint variable* (x, r) to rewrite Eq. (4) as a single-level problem and design an algorithm to solve it.

However, we must ask: is Eq. (4) equivalent to Eq. (1)? It is relatively straightforward to show that their global solutions coincide, using the equivalence between their LL problems. However, just because their global solutions coincide, does not necessarily mean their local solutions coincide. If we are to work with Eq. (4) in place of Eq. (1), then it is critical that the local solutions to Eq. (4) are local solutions to Eq. (1), because Eq. (4) is a non-convex optimization problem. Theoretical analysis of the relationship between the local solutions is substantially more challenging than theoretical analysis of the relationship between the global solutions. Fortunately, we are able to overcome this challenge in Proposition 3.1 of Section 3.1, where we show that local minima of Eq. (4) are also local minima of Eq. (1).

2.2. Single-level DC Reformulation and Algorithm

We now consider solving (4). The value function of the LL problem governed by r is denoted by

$$v(r) := \min \left\{ l(x) \text{ s.t. } P_i(x) \le r_i, \; i = 1, \dots, J \right\}.$$

Throughout the paper, we assume that the following condition holds.

Assumption 2.1. *For each r belonging to any open subset of $\mathbb{R}^J_+$, $\mathcal{F}(r) := \{x \text{ s.t. } P_i(x) \le r_i, i = 1, \dots, J\} \ne \emptyset$ and $l(x)$ is bounded below.*

Thanks to full convexity, $v(r)$ induced by a partial minimization is convex and locally Lipschitz continuous around every point in $\mathbb{R}^J_{>0} := \{r \in \mathbb{R}^J | r_j > 0\}$ (see e.g. Lemma 3 in Ye et al. 2021). Using the value function, we now reformulate BLP (4) as the following DC program:

$$\begin{aligned} \min_{x, r \in \mathbb{R}^J_+} \quad & L(x) \\ \text{s.t.} \quad & l(x) - v(r) \le 0, P_i(x) \le r_i, \; i = 1, \dots, J. \end{aligned} \tag{5}$$

We next propose VF-iDCA to solve BLP (4). By using the main idea of DCA which linearizes the concave part of the DC structure, we propose a sequential convex programming scheme as follows. Given a current iteration (x^k, r^k) for each k, solving the LL problem parameterized by r^k

$$\min_{x} \quad l(x) \text{ s.t. } P_i(x) \le r_i^k, \; i = 1, \dots, J, \tag{6}$$

leads to a solution $\tilde{x}^k \in S_c(r^k)$ and a corresponding Karush-Kuhn-Tucker (KKT) multiplier $\gamma^k \in \mathcal{M}(\tilde{x}^k, r^k)$, where $\mathcal{M}(x, r)$ denotes the set of KKT multipliers of LL problem

$$\begin{aligned} \mathcal{M}(x, r) := \Big\{ & \gamma \in \mathbb{R}^J_+ \mid 0 \in \partial l(x) + \sum_{i=1}^{J} \gamma_i \partial P_i(x), \\ & \gamma_i (P_i(x) - r_i) = 0, i = 1, \dots, J \Big\}. \end{aligned}$$

Then by sensitivity analysis, we have $-\gamma^k \in \partial v(r^k)$; see Proposition 3.3. Compute $z^{k+1} := (x^{k+1}, r^{k+1})$ as an approximate minimizer of the strongly convex subproblem

$$\begin{aligned} \min_{x, r \in \mathbb{R}^J_+} \quad \phi_k(x, r) :=& L(x) + \frac{\rho}{2} \| z - z^k \|^2 \\ & + \alpha_k \max_{i=1,\dots,J} \{ 0, V_k(x, r), P_i(x) - r_i \}, \end{aligned} \tag{7}$$

where $\rho > 0$, and α_k represents the adaptive penalty parameter, $z := (x, r)$, $z^k := (x^k, r^k)$ and

$$V_k(x, r) := l(x) - l(\tilde{x}^k) + \langle \gamma^k, r - r^k \rangle.$$

Denoting $\Sigma := \mathbb{R}^n \times \mathbb{R}^J_+$, we introduce an inexact condition for choosing z^{k+1}:

$$\operatorname{dist}(0, \partial\phi_k(z^{k+1}) + \mathcal{N}_\Sigma(z^{k+1})) \leq \frac{\sqrt{2}}{2}\rho\|z^k - z^{k-1}\|, \quad (8)$$

where $\operatorname{dist}(x, \Omega)$ is the distance from x to Ω and $\mathcal{N}_\Sigma$ denotes the normal cone to Σ. Using above constructions and letting

$$t^{k+1} = \max_{i=1,\dots,J}\{0, V_k(x^{k+1}, r^{k+1}), P_i(x^{k+1}) - r_i^{k+1}\}, \quad (9)$$

we are ready to present VF-iDCA in Algorithm 1.

Algorithm 1 VF-iDCA

```
Take an initial point $(x^0, r^0) \in \mathbb{R}^n \times \mathbb{R}^J_+$; $c_\alpha, \delta_\alpha > 0$;
an initial penalty parameter $\alpha_0 > 0$; tolerance $tol > 0$.
for $k = 0, 1, \dots$ do
  Solve LL problem Eq. (6). Find $\tilde{x}^k \in S_c(r^k)$ and a KKT multiplier $\gamma^k$.
  Solve problem Eq. (7) up to tolerance in (8). Find an approximate solution $z^{k+1} = (x^{k+1}, r^{k+1})$.
  Stopping test: Stop if $\max\{\|z^{k+1} - z^k\|, t^{k+1}\} < tol$.
  Adaptive penalty parameter update: Set
    $\alpha_{k+1} = \begin{cases} \alpha_k + \delta_\alpha, & \text{if } \max\left\{\alpha_k, \frac{1}{t^{k+1}}\right\} < \frac{c_\alpha}{\Delta^{k+1}}, \\ \alpha_k, & \text{otherwise,} \end{cases}$   (10)
  for $t^{k+1}$ defined in (9) and $\Delta^{k+1} := \|z^{k+1} - z^k\|$.
end for
```

Remark 2.1. *When $l(x)$ is differentiable with a Lipschitz continuous gradient, we can derive a linearized version of VF-iDCA where we not only linearize the concave part but also the convex smooth part of the DC structure.*

3. Theoretical Investigations

With the purpose of deriving a provably convergent algorithm for BLP Eq. (1) without LLSC and LLS, we involve two signature features in our algorithmic design, i.e., hyperparameter decoupling and single-level DC reformulation. In this section, we provide convergence analysis of the proposed VF-iDCA towards "good" quality solutions, i.e. towards KKT stationarity solutions, defined as follows.

Definition 3.1. *A feasible point $(\bar{x}, \bar{r})$ of problem Eq. (5) is a KKT stationary solution if there exist multipliers $\eta \geq 0, \lambda \in \mathbb{R}^J_+$ such that*

$$0 \in \partial L(\bar{x}) + \eta\partial l(\bar{x}) + \sum_{i=1}^{J}\lambda_i \partial P_i(\bar{x}),$$

$$\lambda \in -\eta\partial v(\bar{r}) + \mathcal{N}_{\mathbb{R}^J_+}(\bar{r}), \lambda_i\left(P_i(\bar{x}) - \bar{r}_i\right) = 0, i = 1, \dots, J.$$

3.1. Solutions Recovery for Hyperparameter Decoupling

Before analyzing the convergence properties of Algorithm 1, we first need to justify the validity of the hyperparameter decoupling, i.e. we need to show that BLP Eq. (1) is equivalent to BLP Eq. (4).

The following lemmas show a one-to-one relationship between the LL problems Eq. (2) and Eq. (3).

Lemma 3.1. *Let $x \in S_p(\lambda)$ with $\lambda \in \mathbb{R}^J_+$. Then $x \in S_c(r)$ where $r = P(x) \in \mathbb{R}^J_+$.*

Lemma 3.2. *For any $x \in S_c(r)$ with $r \in \mathbb{R}^J_+$, suppose $\lambda \in \mathcal{M}(x, r)$. Then $x \in S_p(\lambda)$.*

Note that the nonemptyness of the multiplier set $\mathcal{M}(x, r)$ can be acheived by imposing certain constraint qualifications such as the Slater condition.

The next theorem shows that the x components of globally optimal solutions for BLPs Eq. (1) and Eq. (4) coincide. We denote by $P(x) := (P_1(x), \dots, P_J(x))$.

Theorem 3.1 (Global solutions recovery)**.** *If $(\bar{x}, \bar{r}) \in \mathbb{R}^n \times \mathbb{R}^J_+$ is a global optimal solution of problem Eq. (4) and $\bar{\lambda} \in \mathcal{M}(\bar{x}, \bar{r})$, then $(\bar{x}, \bar{\lambda})$ is a global optimal solution of BLP Eq. (1). Conversely, suppose there exists a dense subset D of $\mathbb{R}^n \times \mathbb{R}^J_+$ such that $\mathcal{M}(x, r) \neq \emptyset$ for all $(x, r) \in D$. Let $(\bar{x}, \bar{\lambda})$ be a global optimal solution of problem Eq. (1). Then $(\bar{x}, \bar{r})$ with $\bar{r} := P(\bar{x})$ is a global optimal solution of BLP Eq. (4).*

Theorem 3.1 says that the global solutions of the fully convex BLP Eq. (4) are the same as the global solutions of the original BLP Eq. (1).

We now ask an arguably more important question: are the *local* solutions of BLP Eq. (4) the same as the local solutions of BLP Eq. (1)? First, we show in the following proposition that any local optimal solution of BLP Eq. (4) must be locally optimal for BLP Eq. (1).

Proposition 3.1 (Local solution recovery-I)**.** *If $(\bar{x}, \bar{r})$ is a local solution of BLP Eq. (4) with $\bar{r} = P(\bar{x})$ and $\bar{\lambda} \in \mathcal{M}(\bar{x}, \bar{r})$, then $(\bar{x}, \bar{\lambda})$ is a local solution of BLP Eq. (1).*

This is the key result that justifies our choice in this paper to solve the tractable fully convex BLP Eq. (4) in place of the substantially more difficult BLP Eq. (1).

The reverse direction is less practically relevant, because we are unlikely to attempt to solve BLP Eq. (4) by solving the more difficult BLP Eq. (1). Nevertheless, the following result provides conditions under which local solutions to BLP Eq. (1) are local solutions to BLP Eq. (4).

Proposition 3.2 (Local solution recovery-II)**.** *Suppose that there exists a dense subset D of $\mathbb{R}^n \times \mathbb{R}^J_+$ such that $\mathcal{M}(x,r) \neq \emptyset$ for all $(x,r) \in D$. Let $(\bar{x},\bar{\lambda})$ be a local optimal solution of BLP Eq. (1), locally w.r.t. x and globally w.r.t. λ. Then $(\bar{x},\bar{r})$ with $\bar{r} := P(\bar{x})$ is an optimal solution of BLP Eq. (4) locally w.r.t. x and globally w.r.t. r.*

Furthermore, let $(\bar{x},\bar{\lambda})$ be a locally optimal solution of BLP Eq. (1) and $\bar{r} := P(\bar{x})$. If there exist a bounded set Λ and $\epsilon > 0$ such that $\Lambda \cap \mathcal{M}(x,r) \neq \varnothing$ for any $(x,r) \in D \cap \mathbb{B}_\epsilon(\bar{x},\bar{r})$, $\mathcal{M}(\bar{x},\bar{r})$ is a singleton, then $(\bar{x},\bar{r})$ must be locally optimal for BLP Eq. (4).

Remark 3.2. *If both l, P are differentiable, then the existence of dense set and the uniqueness of the multipliers can be simply implied by the linear independence of the gradient vectors $\{\nabla P_i(\bar{x})\}_{i=1}^J$.*

3.2. Sequential Convergence Towards Stationary Solutions

Assumption 3.1. *Throughout this section, we assume the following conditions.*

(a) For all the r^k generated by the algorithm, the lower level solution set $S_c(r^k)$ is nonempty.

(b) For all the r^k and $\tilde{x}^k \in S_c(r^k)$ generated by the algorithm, $\mathcal{M}(\tilde{x}^k, r^k) \neq \emptyset$.

Under Assumption 2.1, by Lemma 3 in Ye et al. (2021), the value function is locally Lipschitz continuous in $\mathbb{R}^J_{>0}$. By Theorem 3 in Ye et al. (2021), we have a characterization for the subdifferential of $v(r)$ in the sense of convex analysis (Rockafellar, 2015).

Proposition 3.3. *For any $r \in \mathbb{R}^J_+$, $x \in S_c(r)$, we have that*

$$\partial v(r) \supseteq -\mathcal{M}(x,r).$$

In addition, if the Slater condition holds for the LL problem, then the equality holds.

The subsequential convergence of VF-iDCA towards stationary solutions follows from Theorem 1 in Ye et al. (2021).

Theorem 3.2. *Suppose that $L(x)$ is bounded below and the sequences $\{z^k\}$ and $\{\alpha_k\}$ generated by the VF-iDCA are bounded. Then any accumulation point $(\bar{x},\bar{r})$ of $\{z^k\}$ is a KKT stationary solution of problem Eq. (5) provided that $\bar{r} \in \mathbb{R}^J_{>0}$.*

Next, as the main contribution of this part, we show that the subsequential convergence can be further enhanced to sequential convergence under the Kurdyka-Łojasiewicz property (Attouch & Bolte, 2009; Attouch et al., 2010; 2013; Bolte et al., 2014). Let $\eta \in [0,+\infty]$ and Φ_η denote the class of all concave and continuous functions $\varphi : [0,\eta) \to [0,+\infty)$ satisfying the conditions: (a) $\varphi(0) = 0$, (b) φ is C^1 on $(0,\eta)$ and continuous at 0, (c) $\varphi'(s) > 0$ for all $s \in (0,\eta)$.

Definition 3.2 (Kurdyka-Łojasiewicz property)**.** *Let $\sigma : \mathbb{R}^d \to (-\infty,+\infty]$ be proper and lower semicontinuous. The function σ is said to have the Kurdyka-Łojasiewicz (KL) property at $\bar{y} \in \operatorname{dom}\partial\sigma := \{y \in \mathbb{R}^d \mid \partial\sigma(y) \neq \varnothing\}$, if there exist $\eta \in (0,+\infty]$, a neighborhood $\mathcal{Y}$ of $\bar{y}$ and a function $\varphi \in \Phi_\eta$, such that for all $y \in \mathcal{Y} \cap \{y \in \mathbb{R}^d \mid \sigma(\bar{y}) < \sigma(y) < \sigma(\bar{y}) + \eta\}$, the following inequality holds*

$$\varphi'(\sigma(y) - \sigma(\bar{y}))\operatorname{dist}(0,\partial\sigma(y)) \geq 1.$$

If σ satisfy the KL property at each point of $\operatorname{dom}\partial\sigma$ then σ is called a KL function.

In addition, given the KL property around any points in a compact set, the uniformized KL property holds; see Lemma 6 in Bolte et al. (2014).

Lemma 3.3 (Uniformized KL property)**.** *Given a compact set C and a proper and lower semicontinuous function $\sigma : \mathbb{R}^d \to (-\infty,+\infty]$, suppose that σ is constant on C and satisfies the KL property at each point of C. Then, there exist ϵ, η and $\varphi \in \Phi_\eta$ such that for all $\bar{y} \in C$ and $y \in \{y \in \mathbb{R}^d \mid \operatorname{dist}(y,C) < \epsilon,\ \ \sigma(\bar{y}) < \sigma(y) < \sigma(\bar{y}) + \eta\}$, it holds*

$$\varphi'(\sigma(y) - \sigma(\bar{y}))\operatorname{dist}(0,\partial\sigma(y)) \geq 1.$$

Inspired by Liu et al. (2019b), we define the following merit function for convergence analysis:

$$\begin{aligned}E_\alpha(z,z_0,\gamma) :=& L(x) + \frac{\rho}{4}\|z - z_0\|^2 + \delta_\Sigma(z)\\ &+ \alpha \max_{i=1,\ldots,J}\{0, l(x) + \langle\gamma, r\rangle + v^*(-\gamma), P_i(x) - r_i\},\end{aligned}$$

where $z := (x,r)$, $z_0 := (x_0,r_0)$, $\delta_\Sigma(z)$ is the indicator function of Σ and v^* denotes the conjugate function of v, that is, $v^*(\xi) := \sup_r\{\langle\xi,r\rangle - v(r)\}$.

Let $\{(x^k,r^k)\}$ be iterates generated by VF-iDCA. Before we can analyze the sequential convergence of $\{(x^k,r^k)\}$, we need the sufficient decrease property and the relative error condition of the merit function E_α, which are summarized in the following lemma.

Lemma 3.4. *Let $\{(x^k,r^k)\}$ be iterates generated by VF-iDCA, then $z^k := (x^k,r^k)$ satisfies*

$$\begin{aligned}&E_{\alpha_k}(z^k,z^{k-1},\gamma^{k-1})\\ &\geq E_{\alpha_k}(z^{k+1},z^k,\gamma^k) + \frac{\rho}{4}\|z^{k+1} - z^k\|^2, \quad \textit{and}\end{aligned} \tag{11}$$

$$\begin{aligned}&\operatorname{dist}\left(0, \partial E_{\alpha_k}(z^{k+1},z^k,\gamma^k)\right)\\ &\leq \frac{\sqrt{2}}{2}\rho\|z^k - z^{k-1}\| + (\alpha_k + \rho)\|z^{k+1} - z^k\|.\end{aligned} \tag{12}$$

Theorem 3.3. *Suppose that $L(x)$ is bounded below and the sequences $\{z^k\}$ and $\{\alpha_k\}$ generated by VF-iDCA are bounded, the merit function E_α is a KL function and there exists $\delta > 0$ such that $r_i^k \geq \delta$ for all k and $i = 1, \dots, J$. Then the sequence $\{z^k\}$ converges to a KKT stationary solution of problem Eq. (5).*

3.3. Further Justification of Validity in Applications

To further justify the validity of our sequential convergence theory in real-world applications, we next discuss the conditions, especially the KL property imposed on the merit function E_α in Theorem 3.3. First, part (b) of Assumption 3.1 holds automatically, for example if the Slater condition holds, i.e., there exists x_0 such that $P_i(x_0) < r_i^k$ for all $i = 1, \dots, J$. Thus, a large class of functions automatically satisfies the KL property, see e.g. Bolte et al. (2010), Attouch et al. (2010), Attouch et al. (2013), and Bolte et al. (2014). As shown in Bolte et al. (2007a) and Bolte et al. (2007b), any semi-algebraic function meets the KL property.

Definition 3.3 (Semi-algebraic sets and functions)**.** *A subset $S \subseteq \mathbb{R}^d$ is called a real semi-algebraic set if there exists a finite number of real polynomial functions $g_{ij}, h_{ij} : \mathbb{R}^d \to \mathbb{R}$ such that*

$$S = \bigcup_{j=1}^{p} \bigcap_{i=1}^{q} \{y \in \mathbb{R}^d \mid g_{ij}(y) = 0,\ h_{ij}(y) < 0\}.$$

A function $h : \mathbb{R}^d \to (-\infty, +\infty]$ is called semi-algebraic if its graph $\{(y, s) \in \mathbb{R}^{d+1} \mid h(y) = s\}$ is a semi-algebraic subset of $\mathbb{R}^{d+1}$.

Lemma 3.5. *Let $\sigma : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and lower semicontinuous. If σ is semi-algebraic then it satisfies the KL property at any point of its domain.*

The class of semi-algebraic functions is closed under various operations, see, e.g., (Attouch et al., 2010; 2013; Bolte et al., 2014). In particular, the indicator functions of semi-algebraic sets, finite sum and product of semi-algebraic functions, composition of semi-algebraic functions and partial minimization of semi-algebraic function over semi-algebraic set are all semi-algebraic functions. Applying this to BLP Eq. (1), when the LL functions $l(x)$ and $P(x)$ are both semi-algebraic, $v(r)$ is a semi-algebraic function. Moreover, if $L(x)$ is a semi-algebraic function, the merit function E_α is also semi-algebraic and hence a KL function.

Theorem 3.4. *Assume that $L(x)$, $l(x)$ and $P(x)$ are semi-algebraic functions. Suppose that $\{z^k := (x^k, r^k)\}$ and $\{\alpha_k\}$ generated by VF-iDCA are bounded, $L(x)$ is bounded below and there exists $\delta > 0$ such that $r_i^k \geq \delta$ for all k and $i = 1, \dots, J$. Then $\{z^k\}$ converges to a KKT point of problem Eq. (5).*

A wide range of functions appearing in applications are semi-algebraic, see, e.g., Attouch et al. (2010), Attouch et al. (2013), and Bolte et al. (2014). In particular, all the functions involved in our applications in the experiments section are semi-algebraic.

4. Extension to General Setting

Our valued function based DC algorithm as well as its convergence analysis can be straightforwardly extended to BLP in a more general setting with LL constraints.

$$\begin{aligned} \min_{x \in X, u \in U, \lambda \in \mathbb{R}_+^J} \quad & L(x, u) \\ \text{s.t.} \quad & x \in \operatorname*{argmin}_{x' \in X} \Big\{ l(x', u) + \sum_{i=1}^{J} \lambda_i P_i(x', u), \\ & \qquad\qquad\qquad \text{s.t.} \;\; g(x', u) \leq 0 \Big\}, \end{aligned} \tag{13}$$

where $X \subseteq \mathbb{R}^n$, $U \subseteq \mathbb{R}^d$ are closed convex sets, $g_i (i = 1, \dots, m), L, l, P_i : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}_+ (i = 1, \dots, J)$ are convex functions defining on an open convex set containing $X \times U$ and $g(x, u) := (g_1(x, u), \dots, g_J(x, u))$. We provide a detailed description of VF-iDCA for BLP Eq. (13) in the Supplemental Material.

5. Experiments

In this section we will compare the performance of our VF-iDCA to widely-used competitors in hyperparameter optimization for a number of machine learning algorithms. Detailed descriptions of these problems are presented in the Supplemental Material. All algorithms were implemented in Python and the software package used for reproduce our experiments is available at `https://github.com/SUSTech-Optimization/VF-iDCA`. The competitors are as follows:

- **Implicit Differentiation:** We consider IGJO in Feng & Simon (2018) and IFDM in Bertrand et al. (2020).
- **Grid Search:** We perform a brute-force grid search over a 10× 10 uniformly-spaced grid.
- **Random Search:** We apply uniform random sampling 100 times at each direction of hyperparameters.
- **TPE:** Tree-structured Parzen Estimator approach (Bergstra et al., 2013) is a Bayesian optimization method based on Gaussian mixture models which can handle a large number of hyperparameters.

All experiments run on a computer with Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz and 16.00 GB memory. We solve the strongly convex subproblem (7) at each iteration of VF-iDCA using the CVXPY package. IGJO and IFDM are implemented using code from `https://github.com/jjfeng/nonsmooth-joint-opt` and `https:`

//github.com/QB3/sparse-ho, respectively. We only use IFDM for elastic net, as its code can only deal with elastic net among our tested problems. Each subproblem in grid search and random search is solved using the CVXPY package. TPE is implemented using code from https://github.com/hyperopt/hyperopt and its subproblem is solved using the CVXPY package. Whenever used, the CVXPY package is applied with the open source solvers ECOS and SCS only.

As for the paramters δ_α and c_α in VF-iDCA, we used $\delta_\alpha = 5$ for all the experiments, and we adopted $c_\alpha = 0.1$ in sparse group lasso while we used $c_\alpha = 1$ for the other applications. When δ_α is too small, the algorithm may converge slower since the algorithm takes more steps to reach the exact penalty. However, in our experience, the performance of Algorithm 1 is not sensitive to the choice of δ_α. The choice of c_α is more important. We tested various ratios for Δ^k and t^k to adjust c_α. When c_α is large, the updating frequency of the penalty parameter will be high, which may slow down the convergence speed.

5.1. Numerical experiments on synthetic data

We consider hyperparameter selection for elastic net and variations on sparse group lasso, low-rank matrix completion, and support vector machines previously explored in Feng & Simon (2018) and Kunapuli et al. (2008); see Section C of the Supplemental Materials for details. Detailed descriptions of the synthetic data generation settings are in Section D of the Supplemental Materials.

5.1.1. Elastic Net

The numerical results on elastic net averaged over 30 data sets are in Table 3. Overall, VF-iDCA attains the highest-quality solutions relatively quickly. In the first two settings, the gradient-based methods (IGJO/IFDM and VF-iDCA) perform better than the gradient-free methods. Furthermore, our VF-iDCA achieves significantly lower validation and test errors than all other methods in the third setting, where $p = 2500$. This suggests that our method is especially competitive when the dimension of the problem is large.

5.1.2. Sparse Group Lasso

Table 4 records the numerical results of sparse group lasso averaged over 30 repetitions. Not only are the solutions of VF-iDCA are better than the gradient free methods, they are also better than the gradient-based IGJO. Furthermore, VF-iDCA is much faster than IGJO.

5.1.3. Low-rank matrix completion

For this problem, we present the numerical results on 60×60 matrices in Table 5.

Table 3. Elastic net problems on synthetic data. Here, $|I_{\text{tr}}|; |I_{\text{val}}|; |I_{\text{test}}|$, and p represent the number of training examples, the number of validation examples, the number of test examples, and the number of features, respectively.

Settings	Method	Time	Val. Err.	Test Err.
$\lvert I_{\text{tr}}\rvert = 100$ $\lvert I_{\text{val}}\rvert = 20$ $\lvert I_{\text{test}}\rvert = 250$ $p = 250$	Grid	3.10 ± 0.44	6.16 ± 2.35	6.68 ± 1.16
	Random	3.55 ± 0.58	5.98 ± 2.24	6.67 ± 1.15
	TPE	5.41 ± 0.75	6.05 ± 2.30	6.77 ± 1.04
	IGJO	2.04 ± 1.46	4.43 ± 1.77	5.13 ± 1.37
	IFDM	1.33 ± 0.55	4.41 ± 0.96	4.77 ± 1.46
	VF-iDCA	**0.91** $\pm$ **0.19**	1.95 ± 0.81	**3.99** $\pm$ **0.69**
$\lvert I_{\text{tr}}\rvert = 100$ $\lvert I_{\text{val}}\rvert = 100$ $\lvert I_{\text{test}}\rvert = 250$ $p = 250$	Grid	3.17 ± 0.43	6.51 ± 1.53	6.82 ± 1.10
	Random	5.29 ± 0.60	6.44 ± 1.53	6.77 ± 1.14
	TPE	5.40 ± 0.84	6.44 ± 1.53	6.76 ± 1.06
	IGJO	2.42 ± 1.30	4.71 ± 1.32	4.88 ± 1.30
	IFDM	1.30 ± 0.41	4.78 ± 1.12	4.61 ± 1.12
	VF-iDCA	**1.37** $\pm$ **0.29**	3.04 ± 0.74	**3.58** $\pm$ **0.60**
$\lvert I_{\text{tr}}\rvert = 100$ $\lvert I_{\text{val}}\rvert = 100$ $\lvert I_{\text{test}}\rvert = 100$ $p = 2500$	Grid	19.05 ± 1.63	7.95 ± 1.10	8.54 ± 0.81
	Random	35.42 ± 3.55	7.90 ± 1.09	8.52 ± 0.79
	TPE	32.17 ± 7.40	7.89 ± 1.11	8.60 ± 0.87
	IGJO	16.12 ± 40.95	7.99 ± 1.18	8.41 ± 0.86
	IFDM	**4.38** $\pm$ **2.53**	7.97 ± 0.83	8.53 ± 1.53
	VF-iDCA	19.97 ± 5.17	1.61 ± 1.85	**5.10** $\pm$ **1.07**

Table 4. Sparse group lasso problems on synthetic data. Recall that $\#\lambda$ is the dimension of hyperparameters, p is the number of features and M is the number of groups. Each group contains p/M features.

Settings	Method	$\#\lambda$	Time	Val. Err.	Test Err.
$p = 600$ $M = 30$	Grid	2	30.38 ± 1.82	42.45 ± 7.67	44.56 ± 7.33
	Random	31	28.54 ± 1.51	39.27 ± 7.32	43.00 ± 8.83
	TPE	31	47.07 ± 4.01	35.69 ± 5.92	40.59 ± 6.67
	IGJO	31	69.62 ± 47.76	30.16 ± 7.41	39.28 ± 6.56
	VF-iDCA	31	**8.13** $\pm$ **1.20**	0.01 ± 0.00	**38.50** $\pm$ **6.00**
$p = 600$ $M = 300$	Grid	2	20.84 ± 1.04	41.88 ± 7.64	44.90 ± 7.02
	Random	301	18.94 ± 1.09	43.92 ± 8.77	47.90 ± 8.55
	TPE	301	76.82 ± 2.55	39.22 ± 6.26	42.93 ± 8.00
	IGJO	301	160.85 ± 71.50	20.37 ± 4.46	38.52 ± 6.78
	VF-iDCA	301	56.73 ± 92.48	19.61 ± 8.33	**33.55** $\pm$ **4.71**
$p = 1200$ $M = 300$	Grid	2	87.20 ± 5.85	49.56 ± 10.76	51.85 ± 12.90
	Random	301	73.75 ± 4.28	53.65 ± 12.03	55.84 ± 14.25
	TPE	301	117.07 ± 5.66	45.94 ± 9.30	51.67 ± 12.29
	IGJO	301	98.35 ± 47.47	20.70 ± 4.70	38.90 ± 7.20
	VF-iDCA	301	**23.41** $\pm$ **1.31**	17.90 ± 3.47	**36.90** $\pm$ **7.48**

5.2. Application to real data

In this part, we will conduct numerical experiments on real datasets. All datasets are from the LIBSVM repository (Chang & Lin, 2011)[1]. For each dataset, we will repeat the experiment 30 times. For each repetition of experiments, we applied a random shuffle to the original dataset before partition, and ran all the algorithms on the same partition.

5.2.1. Elastic Net

This section we will apply our method to choose hyperparameters for the elastic net in real data. We will test on three moderate scale datasets, including gisette (Guyon et al., 2004), duke breast-cancer (West et al., 2001) and sensit (Duarte & Hu, 2004). For datasets gisette, duke breast-

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

Table 5. Low-rank matrix completion problems on synthetic data. #λ is the dimension of hyperparameters.

Method	#λ	Time	Val. Err.	Test Err.
Grid	2	20.67 ± 0.90	0.71 ± 0.21	0.76 ± 0.20
Random	25	32.49 ± 1.84	0.73 ± 0.21	0.80 ± 0.20
TPE	25	35.05 ± 9.37	0.68 ± 0.20	0.76 ± 0.18
IGJO	25	1268.65 ± 365.99	0.68 ± 0.21	0.72 ± 0.18
VF-iDCA	25	51.55 ± 10.43	0.06 ± 0.07	$\mathbf{0.70} \pm \mathbf{0.16}$

cancer, sensit, we randomly extracted 50, 11, 25 examples as training set, respectively; 50, 11, 25 examples as validation set, respectively; and the remaining for testing.

Table 6. Elastic net on madelon, gisette and duke breast-cancer. p, $|I_{\text{tr}}|$, $|I_{\text{val}}|$, $|I_{\text{test}}|$ denote features number, the size of train data, validation data and test data.

Dataset	Method	Time	Val. Err.	Test Err.
gisette $p = 5000$ $\lvert I_{\text{tr}}\rvert = 50$ $\lvert I_{\text{val}}\rvert = 50$ $\lvert I_{\text{test}}\rvert = 5900$	Grid	31.85 ± 3.51	0.22 ± 0.04	0.23 ± 0.01
	Random	55.04 ± 9.05	0.22 ± 0.05	0.23 ± 0.02
	TPE	39.12 ± 6.96	0.22 ± 0.05	0.24 ± 0.02
	IGJO	6.10 ± 3.74	0.24 ± 0.05	0.25 ± 0.03
	IFDM	191.18 ± 202.30	0.22 ± 0.02	0.23 ± 0.03
	VF-iDCA	$\mathbf{5.39} \pm \mathbf{1.62}$	0.00 ± 0.00	$\mathbf{0.19} \pm \mathbf{0.01}$
duke breast-cancer $p = 7129$ $\lvert I_{\text{tr}}\rvert = 11$ $\lvert I_{\text{val}}\rvert = 11$ $\lvert I_{\text{test}}\rvert = 22$	Grid	23.75 ± 4.42	0.29 ± 0.06	0.36 ± 0.08
	Random	39.68 ± 7.89	0.28 ± 0.07	0.41 ± 0.22
	TPE	36.22 ± 22.16	0.27 ± 0.07	0.41 ± 0.24
	IGJO	3.51 ± 2.06	0.30 ± 0.06	0.37 ± 0.08
	IFDM	26.79 ± 73.39	0.32 ± 0.05	0.36 ± 0.08
	VF-iDCA	12.23 ± 15.48	0.00 ± 0.00	$\mathbf{0.29} \pm \mathbf{0.07}$
sensit $p = 78823$ $\lvert I_{\text{tr}}\rvert = 25$ $\lvert I_{\text{val}}\rvert = 25$ $\lvert I_{\text{test}}\rvert = 50$	Grid	1.26 ± 0.11	1.30 ± 0.82	1.31 ± 0.48
	Random	1.29 ± 0.11	1.52 ± 1.15	1.45 ± 0.60
	TPE	1.76 ± 0.09	1.38 ± 0.96	1.39 ± 0.55
	IGJO	0.49 ± 0.74	0.52 ± 0.20	0.61 ± 0.11
	IFDM	6.60 ± 3.49	0.57 ± 0.10	0.61 ± 0.23
	VF-iDCA	$\mathbf{0.16} \pm \mathbf{0.08}$	0.23 ± 0.11	$\mathbf{0.51} \pm \mathbf{0.06}$

5.2.2. Support Vector Machine

We apply our experiments to moderate scale datasets, liver-disorders, diabetes, breast-cancer, sonar, a1a (Asuncion & Newman, 2007), w1a (Platt, 1999). On each dataset, we choose hyperparameters for the SVM via 3/6-fold cross-validation 30 times with the following data partition rule. For a dataset which contains N examples, we extracted $3\lfloor N/6 \rfloor$ examples as the training set used for cross-validation Ω and took the remaining part as the test set Ω_{test}. Since the bilevel program for cross-validated SVM involves LL constraints, IGJO cannot be applied. We implement VF-iDCA with two different tolerances. We denote the one with $tol = 0.01$ by VF-iDCA, and the one with $tol = 0.1$ by VF-iDCA-t. We set $\bar{\mathbf{w}}_{lb} = 10^{-6}$ and $\bar{\mathbf{w}}_{ub} = 10$.

Table 7 and Table 8 record the numerical results of 3-fold SVM and 6-fold SVM on 6 datasets with 30 repeatations, respectively. Fig. 1 shows how the validation error and test error rate changed over time for each algorithm. For such a problem, VF-iDCA shows superiority in both computation time and solution quality. By comparing the numerical results of VF-iDCA and VF-iDCA-t, it can be observed that VF-iDCA can obtain a satisfactory result with a moderate algorithmic tolerance.

The boundedness assumption for the sequence of adaptive penalty parameters, which plays an important role in the convergence analysis, appeared to be satisfied in our extensive experiments. In fact, as discussed in Ye et al. (2021), if we relax the constraint in Eq. (5) to be $l(x) - v(r) \leq \epsilon$ for any small fixed positive ϵ, then the penalty parameters are guaranteed to be bounded. Thus, a small modification of Step 3 of Algorithm 1 would solve this alternative "ϵ-problem", and results analogous to those in Section 3 would hold without assuming boundedness. In the future, we may also try to derive sufficient conditions that directly guarantee the boundedness of the adaptive penalty parameters for Algorithm 1.

Table 7. 3-fold SVM on datasets liver-disorders_scale, diabetes_scale , breast-cancer_scale, sonar, a1a, w1a. #λ is the dimension of hyperparameters.

Dataset	Method	#λ	Time	Val. Err.	Test Err.
liver-disorders_scale $p = 5$ $\lvert\Omega\rvert = 72$ $\lvert\Omega_{\text{test}}\rvert = 73$	Grid	2	0.53 ± 0.01	0.64 ± 0.08	0.34 ± 0.06
	Random	6	0.56 ± 0.03	0.58 ± 0.06	0.32 ± 0.05
	TPE2	2	2.88 ± 1.16	0.61 ± 0.07	0.33 ± 0.06
	TPE	6	0.37 ± 0.29	0.65 ± 0.07	0.34 ± 0.05
	VF-iDCA	6	0.20 ± 0.06	0.53 ± 0.09	0.28 ± 0.05
	VF-iDCA-t	6	$\mathbf{0.09} \pm \mathbf{0.02}$	0.53 ± 0.07	$\mathbf{0.27} \pm \mathbf{0.03}$
diabetes_scale $p = 8$ $\lvert\Omega\rvert = 384$ $\lvert\Omega_{\text{test}}\rvert = 384$	Grid	2	1.70 ± 0.11	0.55 ± 0.03	0.33 ± 0.05
	Random	9	1.83 ± 0.09	0.56 ± 0.04	0.30 ± 0.06
	TPE2	2	18.67 ± 7.84	0.54 ± 0.03	0.32 ± 0.06
	TPE	9	6.64 ± 4.30	0.55 ± 0.03	0.29 ± 0.05
	VF-iDCA	9	0.28 ± 0.03	0.48 ± 0.02	$\mathbf{0.23} \pm \mathbf{0.01}$
	VF-iDCA-t	9	$\mathbf{0.18} \pm \mathbf{0.02}$	0.48 ± 0.02	$\mathbf{0.23} \pm \mathbf{0.01}$
breast-cancer_scale $p = 14$ $\lvert\Omega\rvert = 338$ $\lvert\Omega_{\text{test}}\rvert = 345$	Grid	2	1.63 ± 0.04	0.08 ± 0.01	0.12 ± 0.06
	Random	11	1.80 ± 0.03	0.09 ± 0.01	0.08 ± 0.09
	TPE2	2	14.72 ± 6.02	0.07 ± 0.01	0.09 ± 0.10
	TPE	11	9.14 ± 4.55	0.09 ± 0.01	0.10 ± 0.11
	VF-iDCA	11	1.12 ± 0.59	0.05 ± 0.01	$\mathbf{0.03} \pm \mathbf{0.01}$
	VF-iDCA-t	11	$\mathbf{0.14} \pm \mathbf{0.01}$	0.09 ± 0.01	0.04 ± 0.01
sonar $p = 60$ $\lvert\Omega\rvert = 102$ $\lvert\Omega_{\text{test}}\rvert = 106$	Grid	2	3.19 ± 0.10	0.58 ± 0.08	0.40 ± 0.12
	Random	61	3.23 ± 0.06	0.54 ± 0.06	0.34 ± 0.10
	TPE2	2	18.47 ± 6.84	0.57 ± 0.08	0.37 ± 0.13
	TPE	61	40.77 ± 7.12	0.64 ± 0.10	0.41 ± 0.12
	VF-iDCA	61	2.22 ± 1.50	0.00 ± 0.00	$\mathbf{0.24} \pm \mathbf{0.04}$
	VF-iDCA-t	61	$\mathbf{0.48} \pm \mathbf{0.09}$	0.03 ± 0.02	$\mathbf{0.24} \pm \mathbf{0.04}$
a1a $p = 123$ $\lvert\Omega\rvert = 801$ $\lvert\Omega_{\text{test}}\rvert = 804$	Grid	2	8.04 ± 0.15	0.41 ± 0.02	0.24 ± 0.02
	Random	124	8.62 ± 0.30	0.41 ± 0.02	0.22 ± 0.03
	TPE2	2	65.51 ± 16.24	0.41 ± 0.02	0.24 ± 0.02
	TPE	124	176.59 ± 17.38	0.42 ± 0.03	0.23 ± 0.03
	VF-iDCA	124	10.17 ± 5.47	0.27 ± 0.02	0.18 ± 0.01
	VF-iDCA-t	124	$\mathbf{1.10} \pm \mathbf{0.07}$	0.27 ± 0.02	$\mathbf{0.17} \pm \mathbf{0.01}$
w1a $p = 300$ $\lvert\Omega\rvert = 1236$ $\lvert\Omega_{\text{test}}\rvert = 1241$	Grid	2	20.21 ± 0.82	0.06 ± 0.01	0.03 ± 0.00
	Random	301	20.44 ± 1.10	0.06 ± 0.01	0.03 ± 0.00
	TPE2	2	86.10 ± 28.19	0.06 ± 0.01	0.03 ± 0.00
	TPE	301	299.62 ± 78.72	0.06 ± 0.01	0.03 ± 0.00
	VF-iDCA	301	27.49 ± 7.31	0.01 ± 0.00	$\mathbf{0.02} \pm \mathbf{0.00}$
	VF-iDCA-t	301	$\mathbf{4.87} \pm \mathbf{0.51}$	0.01 ± 0.00	$\mathbf{0.02} \pm \mathbf{0.00}$

6. Conclusion

In this paper, we develop VF-iDCA, a new algorithm for hyperparameter tuning that can be applied for a single training/validation split or for cross-validation. This algorithm heavily exploits the structure of the hyperparameter tuning BLP in order to develop a tailored and effective solution.

Critically, unlike existing gradient-based BLP algorithms, we are able to rigorously establish the first result for sequential convergence towards stationary solutions in the absence of LLSC and LLS simplifications. Furthermore, in our numerical experiments we find that VF-iDCA is able

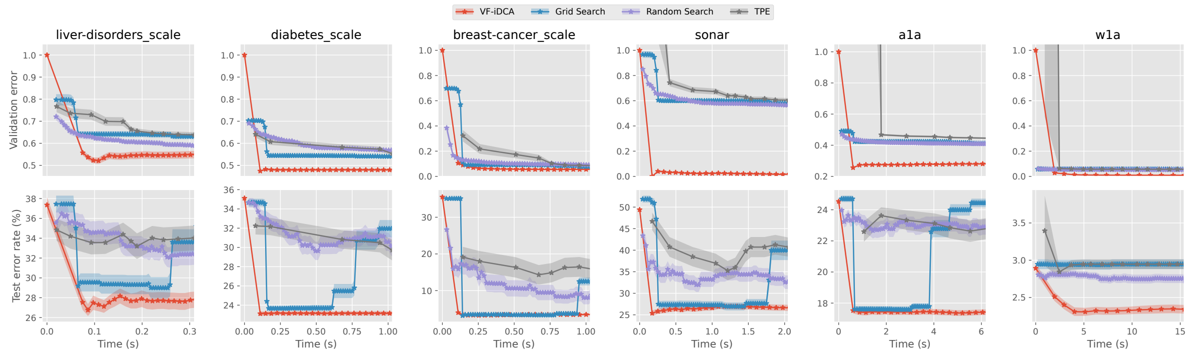

Figure 1. Comparison of the algorithms on SVM problem (validation error and test error versus time) for 6 datasets: liver-disorders_scale, diabetes_scale, breast-cancer_scale, sonar, a1a, w1a

Table 8. 6-fold SVM on datasets liver-disorders_scale, diabetes_scale , breast-cancer_scale, sonar, a1a, w1a. #λ is the dimension of hyperparameters.

Dataset	Method	#λ	Time	Val. Err.	Test Err.
	Grid	2	0.78 ± 0.02	0.63 ± 0.08	0.33 ± 0.07
liver-disorders_scale	Random	6	0.79 ± 0.04	0.62 ± 0.07	0.31 ± 0.05
$p = 5$	TPE2	2	6.88 ± 4.15	0.62 ± 0.07	0.32 ± 0.06
$\|\Omega\| = 72$	TPE	6	1.06 ± 1.04	0.63 ± 0.08	0.34 ± 0.05
$\|\Omega_{\text{test}}\| = 73$	VF-iDCA	6	0.33 ± 0.14	0.41 ± 0.08	$\mathbf{0.27} \pm \mathbf{0.05}$
	VF-iDCA-t	6	$\mathbf{0.18} \pm \mathbf{0.03}$	0.41 ± 0.08	$\mathbf{0.27} \pm \mathbf{0.04}$
	Grid	2	3.18 ± 0.14	0.55 ± 0.03	0.32 ± 0.05
diabetes_scale	Random	9	3.63 ± 0.21	0.56 ± 0.03	0.31 ± 0.05
$p = 8$	TPE2	2	51.85 ± 20.49	0.55 ± 0.03	0.33 ± 0.05
$\|\Omega\| = 384$	TPE	9	29.52 ± 13.13	0.55 ± 0.03	0.27 ± 0.06
$\|\Omega_{\text{test}}\| = 384$	VF-iDCA	9	0.56 ± 0.08	0.43 ± 0.02	$\mathbf{0.23} \pm \mathbf{0.01}$
	VF-iDCA-t	9	$\mathbf{0.35} \pm \mathbf{0.01}$	0.43 ± 0.02	$\mathbf{0.23} \pm \mathbf{0.01}$
	Grid	2	3.38 ± 0.25	0.08 ± 0.02	0.15 ± 0.06
breast-cancer_scale	Random	11	3.92 ± 0.29	0.09 ± 0.02	0.07 ± 0.08
$p = 14$	TPE2	2	38.69 ± 16.27	0.07 ± 0.02	0.08 ± 0.09
$\|\Omega\| = 336$	TPE	11	25.96 ± 12.95	0.09 ± 0.01	0.11 ± 0.13
$\|\Omega_{\text{test}}\| = 347$	VF-iDCA	11	2.01 ± 0.17	0.03 ± 0.01	$\mathbf{0.03} \pm \mathbf{0.01}$
	VF-iDCA-t	11	$\mathbf{0.29} \pm \mathbf{0.08}$	0.08 ± 0.01	$\mathbf{0.04} \pm \mathbf{0.01}$
	Grid	2	6.57 ± 0.32	0.59 ± 0.08	0.39 ± 0.11
sonar	Random	61	6.44 ± 0.28	0.54 ± 0.06	0.32 ± 0.08
$p = 60$	TPE2	2	58.19 ± 29.60	0.57 ± 0.08	0.36 ± 0.12
$\|\Omega\| = 102$	TPE	61	97.65 ± 31.37	0.60 ± 0.07	0.39 ± 0.12
$\|\Omega_{\text{test}}\| = 106$	VF-iDCA	61	0.92 ± 0.02	0.00 ± 0.00	$\mathbf{0.23} \pm \mathbf{0.04}$
	VF-iDCA-t	61	$\mathbf{0.92} \pm \mathbf{0.02}$	0.00 ± 0.00	$\mathbf{0.23} \pm \mathbf{0.04}$
	Grid	2	17.60 ± 0.36	0.40 ± 0.02	0.25 ± 0.02
a1a	Random	124	18.59 ± 0.42	0.40 ± 0.02	0.21 ± 0.03
$p = 123$	TPE2	2	161.68 ± 42.67	0.40 ± 0.02	0.24 ± 0.02
$\|\Omega\| = 798$	TPE	124	312.63 ± 60.60	0.41 ± 0.03	0.23 ± 0.03
$\|\Omega_{\text{test}}\| = 807$	VF-iDCA	124	63.01 ± 186.14	0.18 ± 0.02	$\mathbf{0.17} \pm \mathbf{0.01}$
	VF-iDCA-t	124	$\mathbf{4.22} \pm \mathbf{0.37}$	0.19 ± 0.01	$\mathbf{0.17} \pm \mathbf{0.01}$
	Grid	2	44.29 ± 1.39	0.05 ± 0.00	0.03 ± 0.00
w1a	Random	301	61.80 ± 2.91	0.05 ± 0.00	0.03 ± 0.00
$p = 300$	TPE2	2	190.04 ± 39.00	0.05 ± 0.00	0.03 ± 0.00
$\|\Omega\| = 1236$	TPE	301	703.72 ± 82.75	0.05 ± 0.01	0.03 ± 0.00
$\|\Omega_{\text{test}}\| = 1241$	VF-iDCA	301	97.50 ± 35.99	0.01 ± 0.00	$\mathbf{0.02} \pm \mathbf{0.00}$
	VF-iDCA-t	301	$\mathbf{26.74} \pm \mathbf{3.67}$	0.01 ± 0.00	$\mathbf{0.02} \pm \mathbf{0.00}$

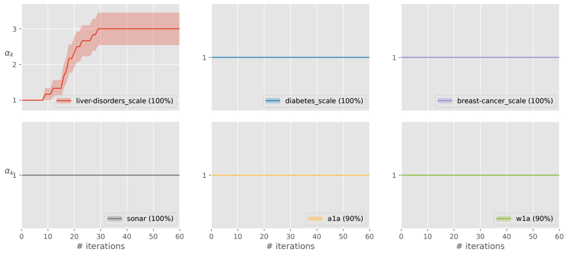

Figure 2. Adaptive penalty parameter α_k (defined in Eq. (10)) of the VF-iDCA on SVM problem for 6 datasets: liver-disorders_scale, diabetes_scale, breast-cancer_scale, sonar, a1a, w1a

to match or outperform the state-of-the-art hyperparameter tuning approaches in terms of validation and test error for a number of machine learning algorithms, especially when the number of hyperparameters is large. In many cases, VF-iDCA also enjoys considerable gains in computational efficiency compared to the state-of-the-art.

The boundedness assumption of the sequence of adaptive penalty parameters, which plays an important role in the convergence analysis, appeared to be satisfied in our extensive experiments. In fact, as discussed in Ye et al. (2021), if we relax the constraint in Eq. (5) to be $l(x) - v(r) \leq \epsilon$ for any small fixed positive ϵ, then the penalty parameters are guaranteed to be bounded. Thus, a small modification of Step 3 of Algorithm 1 would solve this alternative "ϵ-problem", and results analogous to those in Section 3 would hold without assuming boundedness. In the future, we may also try to derive sufficient conditions that directly guarantee the boundedness of the adaptive penalty parameters for Algorithm 1.

Acknowledgements

The alphabetical order of the authors indicates the equal contribution to the paper. This work is partially supported by NSERC, the National Natural Science Foundation of China (No. 11971220), the Shenzhen Science and Technology Program (No. RCYX20200714114700072), the Guangdong Basic and Applied Basic Research Foundation (No. 2022B1515020082) and the Pacific Institute for the Mathematical Sciences (PIMS).

References

Asuncion, A. and Newman, D. UCI machine learning repository, 2007.

Attouch, H. and Bolte, J. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1):5–16, 2009.

Attouch, H., Bolte, J., Redont, P., and Soubeyran, A. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

Attouch, H., Bolte, J., and Svaiter, B. F. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1):91–129, 2013.

Bergstra, J., Yamins, D., and Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *ICML*, pp. 115–123. PMLR, 2013.

Bertrand, Q., Klopfenstein, Q., Blondel, M., Vaiter, S., Gramfort, A., and Salmon, J. Implicit differentiation of lasso-type models for hyperparameter optimization. In *ICML*, pp. 810–821. PMLR, 2020.

Bertrand, Q., Klopfenstein, Q., Massias, M., Blondel, M., Vaiter, S., Gramfort, A., and Salmon, J. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *arXiv preprint arXiv:2105.01637*, 2021.

Bolte, J., Daniilidis, A., and Lewis, A. The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007a.

Bolte, J., Daniilidis, A., Lewis, A., and Shiota, M. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007b.

Bolte, J., Daniilidis, A., Ley, O., and Mazet, L. Characterizations of łojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

Bolte, J., Sabach, S., and Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.

Chang, C.-C. and Lin, C.-J. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

Clarke, F. H. *Optimization and nonsmooth analysis*. SIAM, 1990.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Duarte, M. F. and Hu, Y. H. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.

Feng, J. and Simon, N. Gradient-based regularization parameter selection for problems with nonsmooth penalty functions. *Journal of Computational and Graphical Statistics*, 27(2):426–435, 2018.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pp. 1126–1135. PMLR, 2017.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. Forward and reverse gradient-based hyperparameter optimization. In *ICML*, pp. 1165–1173. PMLR, 2017.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, pp. 1563–1572. PMLR, 2018.

Grazzi, R., Franceschi, L., Pontil, M., and Salzo, S. On the iteration complexity of hypergradient computation. In *ICML*, pp. 3748–3758. PMLR, 2020.

Guyon, I., Gunn, S., Ben-Hur, A., and Dror, G. Result analysis of the nips 2003 feature selection challenge. *NeurIPS*, 17, 2004.

Ji, K. and Liang, Y. Lower bounds and accelerated algorithms for bilevel optimization. *arXiv preprint arXiv:2102.03926v2*, 2021.

Ji, K., Lee, J. D., Liang, Y., and Poor, H. V. Convergence of meta-learning with task-specific adaptation over partial parameters. *NeurIPS*, 33:11490–11500, 2020a.

Ji, K., Yang, J., and Liang, Y. Bilevel optimization: Nonasymptotic analysis and faster algorithms. *arXiv preprint arXiv:2010.07962v2*, 2020b.

Ji, K., Yang, J., and Liang, Y. Bilevel optimization: Convergence analysis and enhanced design. In *ICML*, pp. 4882–4892. PMLR, 2021.

Ji, K., Liu, M., Liang, Y., and Ying, L. Will bilevel optimizers benefit from loops. *arXiv preprint arXiv:2205.14224*, 2022.

Kunapuli, G. *A bilevel optimization approach to machine learning*. Rensselaer Polytechnic Institute, 2008.

Kunapuli, G., Bennett, K. P., Hu, J., and Pang, J.-S. Classification model selection via bilevel programming. *Optimization Methods & Software*, 23(4):475–489, 2008.

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. In *ICLR*, 2019a.

Liu, R., Mu, P., Yuan, X., Zeng, S., and Zhang, J. A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In *ICML*, pp. 6305–6315. PMLR, 2020.

Liu, R., Liu, X., Yuan, X., Zeng, S., and Zhang, J. A value-function-based interior-point method for non-convex bi-level optimization. In *ICML*, pp. 6882–6892. PMLR, 2021a.

Liu, R., Liu, Y., Zeng, S., and Zhang, J. Towards gradient-based bilevel optimization with non-convex followers and beyond. *NeurIPS*, 34, 2021b.

Liu, R., Mu, P., Yuan, X., Zeng, S., and Zhang, J. A general descent aggregation framework for gradient-based bi-level optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Liu, T., Pong, T. K., and Takeda, A. A refined convergence analysis of pDCA$_e$ pdca e with applications to simultaneous sparse recovery and outlier detection. *Computational Optimization and Applications*, 73(1):69–100, 2019b.

Lorraine, J. and Duvenaud, D. Stochastic hyperparameter optimization through hypernetworks. *CoRR, abs/1802.09419*, 2018.

Lorraine, J., Vicol, P., and Duvenaud, D. Optimizing millions of hyperparameters by implicit differentiation. In *AISTATS*, pp. 1540–1552. PMLR, 2020.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *CoRR, abs/1803.02999*, 2018.

Pedregosa, F. Hyperparameter optimization with approximate gradient. In *ICML*, pp. 737–746. PMLR, 2016.

Platt, J. C. Fast training of support vector machines using sequential minimal optimization, advances in kernel methods. 1999.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. *NeurIPS*, 32, 2019.

Rockafellar, R. T. *Convex analysis*. Princeton university press, 2015.

Shaban, A., Cheng, C., Hatch, N., and Boots, B. Truncated back-propagation for bilevel optimization. In *AISTATS*, 2019.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

Sow, D., Ji, K., Guan, Z., and Liang, Y. A constrained optimization approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*, 2022.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J. A., Marks, J. R., and Nevins, J. R. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences*, 98(20):11462–11467, 2001.

Ye, J. J., Yuan, X., Zeng, S., and Zhang, J. Difference of convex algorithms for bilevel programs with applications in hyperparameter selection. *arXiv preprint arXiv:2102.09006*, 2021.

Zou, H. and Hastie, T. Regression shrinkage and selection via the elastic net, with applications to microarrays. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67:301–20, 2003.

The supplemental materials are organized as follows. In Appendix A, we present all the detailed proofs of the theoretical results given in Section 3. And in Appendix B, the detailed description of the VF-iDCA for bilevel problem in general setting is presented. Appendix C is devoted to the illustration of hyperparameter decoupling for BLP applications. Additional information on how the numerical experiments were run are given in Appendix D.

A. Detailed Proofs

A.1. Proof of Lemma 3.1

By convexity, x is an optimal solution for problem Eq. (2) if and only if its first-order optimality condition

$$0 \in \partial l(x) + \sum_{i=1}^{J} \lambda_i \partial P_i(x) \tag{14}$$

holds. Assuming the KKT condition for problem Eq. (3)

$$\begin{aligned} \exists \lambda \in \mathbb{R}^J_+ \text{ s.t. } 0 \in \partial l(x) + \sum_{i=1}^{J} \lambda_i \partial P_i(x), \\ \lambda_i (P_i(x) - r_i) = 0, i = 1, \ldots, J, \\ P_i(x) - r_i \leq 0, i = 1, \ldots, J, \end{aligned} \tag{15}$$

holds, then $x \in S_c(r)$. Since the difference between the KKT conditions Eq. (14) and Eq. (15) is the complementary slackness conditions $\lambda_i(P_i(x) - r_i) = 0, P_i(x) - r_i \leq 0, i = 1, \ldots, J$, which holds automatically if $P(x) = r$, the conclusion follows immediately.

A.2. Proof of Lemma 3.2

Since $\lambda \in \mathcal{M}(x, r)$, we have

$$0 \in \partial l(x) + \sum_{i=1}^{J} \lambda_i \partial P_i(x).$$

By convexity we have $x \in S_p(\lambda)$.

A.3. Proof of Theorem 3.1

Problems Eq. (1) and Eq. (4) can be equivalently rewritten in the following form

$$\begin{aligned} \min \quad & L(x) \\ \text{s.t.} \quad & x \in S_p(\lambda), \lambda \in \mathbb{R}^J_+, \end{aligned} \tag{16}$$

and

$$\begin{aligned} \min \quad & L(x) \\ \text{s.t.} \quad & x \in S_c(r), r \in \mathbb{R}^J_+, \end{aligned} \tag{17}$$

respectively.

(1) Suppose that $(\bar{x}, \bar{r}) \in \mathbb{R}^n \times \mathbb{R}^J_+$ is a global optimal solution of problem Eq. (17) and $\bar{\lambda} \in \mathcal{M}(\bar{x}, \bar{r})$. We want to show that $(\bar{x}, \bar{\lambda})$ is a global optimal solution of problem Eq. (16). Let (x, λ) be a feasible solution of problem Eq. (16) which means that $x \in S_p(\lambda), \lambda \in \mathbb{R}^J_+$. By Lemma 3.1, $x \in S_c(r)$ with $r = P(x) \in \mathbb{R}^J_+$ and hence (x, r) is a feasible solution of problem Eq. (17). Hence by the optimality of $(\bar{x}, \bar{r})$ to problem Eq. (17), we have $L(\bar{x}) \leq L(x)$. This shows that $(\bar{x}, \bar{\lambda})$ is a global optimal solution of problem Eq. (16).

(2) Suppose that $(\bar{x}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^J_+$ is a global optimal solution of problem Eq. (16). We want to show that $(\bar{x}, \bar{r})$ with $\bar{r} := P(\bar{x})$ is a global optimal solution of problem Eq. (17). Since $\bar{x} \in S_p(\bar{\lambda})$, by Lemma 3.1, $\bar{x} \in S_c(\bar{r})$ with $\bar{r} = P(\bar{x})$. Hence $(\bar{x}, \bar{r})$ is a feasible solution of problem Eq. (17). Let (x, r) be a feasible solution of problem Eq. (17) which means that $x \in S_c(r), r \in \mathbb{R}^J_+$. As D is a dense subset of $X \times \mathbb{R}^J_+$, there exists a sequence $\{(x^k, r^k)\} \subseteq D$ satisfying $(x^k, r^k) \to (x, r)$. Since $\mathcal{M}(x^k, r^k) \neq \emptyset$ for all $(x^k, r^k) \in D$, we can pick $\lambda^k \in \mathcal{M}(x^k, r^k)$. By Lemma 3.2, $x^k \in S_p(\lambda^k)$ and hence (x^k, λ^k) is a feasible solution of problem Eq. (16). By the optimality of $(\bar{x}, \bar{\lambda})$ to problem Eq. (16), we have $L(\bar{x}) \leq L(x^k)$. By taking $k \to \infty$ and the continuity of L, this shows that $(\bar{x}, \bar{r})$ is a global optimal solution of problem Eq. (17).

A.4. Proof of Proposition 3.1

Since $(\bar{x}, \bar{r})$ is a local optimal solution of problem Eq. (17), there is a $\epsilon_0 > 0$ such that $\forall x \in S_c(r), r \in \mathbb{R}^J_+$, and $(x, r) \in \mathbb{B}_{\epsilon_0}(\bar{x}, \bar{r})$,

$$L(\bar{x}) \leq L(x), \tag{18}$$

where $\mathbb{B}_{\epsilon_0}(\bar{x}, \bar{r})$ denotes the closed ball centered at $(\bar{x}, \bar{r})$ with radius ϵ_0. Let (x, λ) be a feasible solution of problem Eq. (16) which means that $x \in S_p(\lambda), \lambda \in \mathbb{R}^J_+$. By Lemma 3.1, $x \in S_c(r)$ with $r = P(x) \in \mathbb{R}^J_+$ and hence (x, r) is a feasible solution of problem Eq. (17). Moreover since P is continuous, we can find $0 < \epsilon_1 < \epsilon_0$ such that as $x \in \mathbb{B}_{\epsilon_1}(\bar{x})$, $r = P(x) \in \mathbb{B}_{\epsilon_0}(\bar{r})$. Hence by Eq. (18), the local optimality of $(\bar{x}, \bar{r})$ to problem Eq. (17), we have $L(\bar{x}) \leq L(x)$. This shows that $(\bar{x}, \bar{\lambda})$ is a local optimal solution of problem Eq. (16).

A.5. Proof of Proposition 3.2

Since $(\bar{x}, \bar{\lambda})$ is a local optimal solution of problem Eq. (16) locally with respect to variable x and globally with respect to variable λ, there is a $\epsilon_0 > 0$ such that $\forall x \in S_p(\lambda), \lambda \in \mathbb{R}^J_+$, and $x \in \mathbb{B}_{\epsilon_0}(\bar{x})$,

$$L(\bar{x}) \leq L(x). \tag{19}$$

Since $\bar{x} \in S_p(\bar{\lambda})$, by Lemma 3.1, $\bar{x} \in S_c(\bar{r})$ with $\bar{r} = P(\bar{x})$. Hence $(\bar{x}, \bar{r})$ is a feasible solution of problem Eq. (17). Let (x, r) be a feasible solution of problem Eq. (17)) which means that $x \in S_c(r), r \in \mathbb{R}^J_+$. As D is a dense subset of $X \times \mathbb{R}^J_+$, there exists a sequence $\{(x^k, r^k)\} \subseteq D$ satisfying $(x^k, r^k) \to (x, r)$. Since $\mathcal{M}(x^k, r^k) \neq \emptyset$ for all $(x^k, r^k) \in D$, we can pick $\lambda^k \in \mathcal{M}(x^k, r^k)$. By Lemma 3.2, $x^k \in S_p(\lambda^k)$ and hence (x^k, λ^k) is a feasible solution of problem Eq. (16). Moreover suppose that x and $\{x^k\}$ lie in $\mathbb{B}_{\epsilon_0}(\bar{x}, \bar{u})$. Hence by (19), we have $L(\bar{x}) \leq L(x^k)$. Taking $k \to \infty$ implies $L(\bar{x}) \leq L(x)$. This shows that $(\bar{x}, \bar{r})$ is a local optimal solution of problem Eq. (17) locally with respect to variable x and globally with respect to variable r.

Now suppose that $\mathcal{M}(\bar{x}, \bar{r}) = \{\bar{\lambda}\}$ is a singleton. Let $(\bar{x}, \bar{\lambda})$ be a local optimal solution of problem Eq. (16). Then there is a $\epsilon_1 > 0$ such that $\forall x \in S_p(\lambda), \lambda \in \mathbb{R}^J_+$ and $(x, \lambda) \in \mathbb{B}_{\epsilon_1}(\bar{x}, \bar{\lambda})$,

$$L(\bar{x}) \leq L(x). \tag{20}$$

Let (x, r) be a feasible solution of problem Eq. (17) which lies in $\mathbb{B}_{\epsilon_2}(\bar{x}, \bar{r})$ for some $0 < \epsilon_2 < \epsilon_1$. It implies that $x \in S_c(r), r \in \mathbb{R}^J_+$ and $(x, r) \in \mathbb{B}_{\epsilon_2}(\bar{x}, \bar{r})$. By the assumption of the uniqueness of the multipliers, we can show through proof by contradiction that when ϵ_2 is sufficiently small, for $x \in \mathbb{B}_{\epsilon_2}(\bar{x})$, $\Lambda \cap \mathcal{M}(x, r) \subseteq \mathbb{B}_{\epsilon_1}(\bar{\lambda})$. We can find a sequence $\{(x^k, r^k)\} \subseteq D \cap \mathbb{B}_{\epsilon}(\bar{x}, \bar{r})$ satisfying $(x^k, r^k) \to (x, r)$. Since $\Lambda \cap \mathcal{M}(x^k, r^k) \neq \emptyset$ for all $(x^k, r^k) \in D \cap \mathbb{B}_{\epsilon}(\bar{x}, \bar{r})$, we can pick $\lambda^k \in \Lambda \cap \mathcal{M}(x^k, r^k)$. We can assume without loss of generality that $(x^k) \in \mathbb{B}_{\epsilon_2}(\bar{x})$ and thus $\lambda^k \in \mathbb{B}_{\epsilon_1}(\bar{\lambda})$. By Lemma 3.2, $x^k \in S_p(\lambda^k)$ and hence (x^k, λ^k) is a feasible solution of problem Eq. (16). Hence by Eq. (20), we have $L(\bar{x}) \leq L(x_k)$. Taking $k \to \infty$ gives us $L(\bar{x}) \leq L(x)$. This shows that $(\bar{x}, \bar{r})$ is a local optimal solution of problem Eq. (17).

A.6. Proof of Lemma 3.4

For convenience, in the proof we denote by $g_k(z) := \max_{i=1,\ldots,J}\{0, V_k(z), P_i(x) - r_i\}$. Then

$$\phi_k(z) = L(x) + \frac{\rho}{2}\|z - z^k\|^2 + \alpha_k g_k(z), \tag{21}$$

where $V_k(z) = l(x) - v(r^k) + \langle \gamma^k, r - r^k \rangle$. By using the same arguments as in the proof of Lemma 1 in (Ye et al., 2021), we can have following decrease result on ϕ_k,

$$\phi_k(z^{k+1}) \leq \phi_k(z^k) + \frac{\rho}{4}\|z^k - z^{k-1}\|^2. \tag{22}$$

Next since v^* is the conjugate function of v, by definition

$$-v(r^k) \leq \langle \gamma^{k-1}, r^k \rangle + v^*(-\gamma^{k-1}). \tag{23}$$

Denote by

$$\tilde{g}(z, \gamma) := \max_{i=1,\ldots,J}\{0, l(x) + \langle \gamma, r \rangle + v^*(-\gamma), P_i(x) - r_i\}.$$

Then

$$E_\alpha(z, z_0, \gamma) := L(x) + \frac{\rho}{4}\|z - z_0\|^2 + \delta_\Sigma(z) + \alpha_k \tilde{g}(z, \gamma).$$

Combining Eq. (21), Eq. (22) and Eq. (23), and taking into account the fact that $z^k \in \Sigma$, we obtain

$$\phi_k(z^{k+1}) \leq E_{\alpha_k}(z^k, z^{k-1}, \gamma^{k-1}). \tag{24}$$

Again, as v^* is the conjugate function of v, and $-\gamma^k \in \partial v(r^k)$, there holds that

$$-v(r^k) - \langle \gamma^k, r^k \rangle = v^*(-\gamma^k),$$

which implies

$$\phi_k(z^{k+1}) = E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) + \frac{\rho}{4}\|z^{k+1} - z^k\|^2. \tag{25}$$

Combining with Eq. (22) and Eq. (24) gives us Eq. (11).

It remains to prove Eq. (12). Since $l(x)$ and $P_i(x) - r_i$, $i = 1, \ldots, J$ are all convex and continuous and thus regular, by the calculus rule for the pointwise maximum (see e.g. Proposition 2.3.12 in (Clarke, 1990)), we have that $g_k(z)$ is regular and

$$\begin{aligned}
&\partial g_k(z) = \\
&\left\{ \eta(\partial l(x) \times \{\gamma^k\}) + \sum_{i=1}^{J} \lambda_i \partial P_i(x) \times \{-\lambda_i e_i\} \right. \\
&\quad \text{s.t. } \lambda_i \in [0,1], \lambda_i\left(P_i(x) - r_i - g_k(z)\right) = 0, \\
&\quad \eta \in [0,1], \eta(l(x) - v(r^k) + \langle \gamma^k, r - r^k \rangle - g_k(z)) = 0, \\
&\quad \left. g_k(z)(1 - \eta - \sum_{i=1}^{J} \lambda_i) = 0, \eta + \sum_{i=1}^{J} \lambda_i \leq 1 \right\},
\end{aligned} \tag{26}$$

where e_i denotes the unit vector with the ith component equal to 1.

Similarly, $\tilde{g}(z, \gamma)$ is regular and

$$\begin{aligned}
&\partial_z \tilde{g}(z, \gamma^k) = \\
&\left\{ \eta(\partial l(x) \times \{\gamma^k\}) + \sum_{i=1}^{J} \lambda_i \partial P_i(x) \times \{-\lambda_i e_i\} \right. \\
&\quad \text{s.t. } \lambda_i \in [0,1], \lambda_i\left(P_i(x) - r_i - \tilde{g}(z, \gamma^k)\right) = 0, \\
&\quad \eta \in [0,1], \eta(l(x) + \langle \gamma^k, r \rangle + v^*(-\gamma^k) - \tilde{g}(z, \gamma^k)) = 0, \\
&\quad \left. \tilde{g}(z, \gamma^k)(1 - \eta - \sum_{i=1}^{J} \lambda_i) = 0, \eta + \sum_{i=1}^{J} \lambda_i \leq 1 \right\}.
\end{aligned} \tag{27}$$

We also have

$$\begin{aligned}
&\partial_\gamma \tilde{g}(z^{k+1}, \gamma) = \left\{ \eta(r^{k+1} - \partial v^*(-\gamma)) \text{ s.t. } \eta \in [0,1], \right. \\
&\quad \left. \eta\left(l(x) + \langle \gamma, r^{k+1} \rangle + v^*(-\gamma) - \tilde{g}(z^{k+1}, \gamma)\right) = 0 \right\}.
\end{aligned} \tag{28}$$

Since v^* is the conjugate function of v, and $-\gamma^k \in \partial v(r^k)$, there holds that

$$-v(r^k) - \langle \gamma^k, r^k \rangle = v^*(-\gamma^k).$$

Hence

$$\begin{aligned} V_k(z^{k+1}) &= l(x^{k+1}) - v(r^k) + \langle \gamma^k, r^{k+1} - r^k \rangle \\ &= l(x^{k+1}) + \langle \gamma^k, r^{k+1} \rangle + v^*(-\gamma^k), \end{aligned}$$

and thus

$$g_k(z^{k+1}) = \tilde{g}(z^{k+1}, \gamma^k),$$

and

$$\partial g_k(z^{k+1}) = \partial_z \tilde{g}(z^{k+1}, \gamma^k). \tag{29}$$

Since $L(x)$ and $g_k(z)$ are all convex and continuous and thus regular, according to the the subdifferential sum rule we have (see e.g. Proposition 2.3.3 in Clarke (1990)), we have

$$\partial \phi_k(z) = \partial L(x) \times \{0\} + \rho(z - z^k) + \alpha_k \partial g_k(z).$$

Similarly we have

$$\begin{aligned} \partial_z E_{\alpha_k}(z, z^k, \gamma^k) =& \partial L(x) \times \{0\} + \frac{\rho}{2}(z - z^k) \\ &+ \mathcal{N}_\Sigma(z^{k+1}) + \alpha_k \partial_z \tilde{g}(z, \gamma^k). \end{aligned}$$

Hence by Eq. (29), we have

$$\begin{aligned} &\partial_z E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) \\ =& \partial \phi_k(z^{k+1}) + \mathcal{N}_\Sigma(z^{k+1}) - \frac{\rho}{2}(z^{k+1} - z^k). \end{aligned} \tag{30}$$

By the partial subdifferential formula (see e.g. by Proposition 2 in (Ye et al., 2021)) and Eq. (28), there exists $\eta_{k+1} \in [0, 1]$ such that

$$\begin{aligned} & \partial E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) \\ = \ & \partial_z E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) \times \partial_{z_0} E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) \\ & \times \partial_\gamma E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) \\ = \ & \partial_z E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) \times \{\frac{\rho}{2}(z^k - z^{k+1})\} \\ & \times \partial_\gamma \tilde{g}(z^{k+1}, \gamma^k) \\ \supseteq \ & \partial_z E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) \times \{\frac{\rho}{2}(z^k - z^{k+1})\} \\ & \times \{\alpha_k \eta_{k+1}(r^{k+1} - \partial v^*(-\gamma^k))\}. \end{aligned}$$

Since z^{k+1} is an approximate solution to problem Eq. (5) satisfying inexact criteria Eq. (8), there exists a vector e_k such that

$$e_k \in \partial \phi_k(z^{k+1}) + \mathcal{N}_\Sigma(z^{k+1})$$

satisfying

$$\|e_k\| \le \frac{\sqrt{2}}{2} \rho \|z^k - z^{k-1}\|.$$

It follows from Eq. (30) that

$$e_k - \frac{\rho}{2}(z^{k+1} - z^k) \in \partial_z E_{\alpha_k}(z^{k+1}, z^k, \gamma^k).$$

Next, using the fact that $-\gamma^k \in \partial v(r^k)$ which is equivalent to $r^k \in \partial v^*(-\gamma^k)$, we have

$$\partial E_{\alpha_k}(z^{k+1}, z^k, \gamma^k) \ni \begin{pmatrix} e_k - \frac{\rho}{2}(z^{k+1} - z^k) \\ \frac{\rho}{2}(z^k - z^{k+1}) \\ \alpha_k \eta_{k+1}(r^{k+1} - r^k) \end{pmatrix}.$$

Eq. (12) then follows immediately.

A.7. Proof of Theorem 3.3

By the assumption that the adaptive penalty sequence $\{\alpha_k\}$ is bounded, $\alpha_k = \bar{\alpha}$ for all sufficiently large k and we can assume without loss of generality that $\alpha_k = \bar{\alpha}$ for all k. Since $L(x)$ is assumed to be bounded below, $E_{\bar{\alpha}}(z, z_0, \gamma)$ is also bounded below. Then, according to Lemma 3.4, we have that $\lim_{k\to\infty} \|z^{k+1} - z^k\|^2 = 0$.

Let C denote the set of all limit points of the sequence $\{(z^k, z^{k-1}, \gamma^{k-1})\}$. Then we have that C is a closed set and $\lim_{k\to\infty} \mathrm{dist}((z^k, z^{k-1}, \gamma^{k-1}), C) = 0$. And by assumption that there exists $\delta > 0$ such that $r_i^k \ge \delta$ for all k and all $i = 1, \ldots, J$ and Theorem 3.2, any accumulation point of the sequence $\{z^k\}$ corresponds to those in the set C is a KKT stationary point of problem Eq. (5). According to Assumption 2.1, the value function is locally Lipschitz continuous around any point in $\mathbb{R}^J_{>0}$. Then by the assumptions that sequence $\{z^k\}$ is bounded, $r_i^k \ge \delta$ for some $\delta > 0$ and $-\gamma^{k-1} \in \partial v(z^{k-1})$, we get the boundedness of the sequence $\{\gamma^k\}$ and thus C is a compact set.

Since by Lemma 3.4, $E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1})$ is decreasing, $E_{\bar{\alpha}}(z, z_0, \gamma)$ is bounded below and continuous on Σ and $\lim_{k\to\infty} \|z^{k+1} - z^k\| = 0$, we have that for any subsequence $\{(z^l, z^{l-1}, \gamma^{l-1})\}$ of the sequence $\{(z^k, z^{k-1}, \gamma^{k-1})\}$,

$$\bar{E} = \lim_{l\to\infty} E_{\bar{\alpha}}(z^l, z^{l-1}, \gamma^{l-1}) = \lim_{k\to\infty} E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}),$$

and thus $E_{\bar{\alpha}}$ is constant on C. We can assume that $E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) > \bar{E}$ for all k. Otherwise, if there exists $k > 0$ such that $E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) = \bar{E}$, then by the above equation and Lemma 3.4, we get $\|z^{k+1} - z^k\| = 0$ when k is sufficiently large, which implies the convergence of sequence $\{z^k\}$ and the conclusion follows immediately.

Since $\lim_{k\to\infty} \mathrm{dist}((z^k, z^{k-1}, \gamma^{k-1}), C) = 0$ and $\lim_{k\to\infty} E_{\bar{\alpha}}(z^k, z^{k-1}, \zeta^{k-1}) = \bar{E}$, for any $\epsilon, \eta > 0$, there exists k_0 such that $\mathrm{dist}((z^k, z^{k-1}, \zeta^{k-1}), C) < \epsilon$ and $\bar{E} < E_{\bar{\alpha}}(z^k, z^{k-1}, \zeta^{k-1}) < \bar{E} + \eta$ for $k \ge k_0$. Since $E_{\bar{\alpha}}(z, z_0, \gamma)$ satisfies the Kurdyka-Łojasiewicz property at each point in C, and $E_{\bar{\alpha}}$ is a finite constant on C, we can apply Lemma 3.3 to obtain a continuous concave function φ such that for any $k \ge k_0$,

$$\begin{aligned} &\varphi'(E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) - \bar{E}) \mathrm{dist}\left(0, \partial E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1})\right) \\ &\ge 1. \end{aligned}$$

Combining with Eq. (12) yields

$$\varphi'\left(E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) - \bar{E}\right) \cdot \left(\frac{\sqrt{2}}{2}\rho\|z^{k-1} - z^{k-2}\| + (\bar{\alpha}+\rho)\|z^k - z^{k-1}\|\right) \geq 1.$$

The concavity of φ and Eq. (11) implies

$$\begin{aligned}
&\varphi'(E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) - \bar{E}) \cdot \frac{\rho}{4}\|z^{k+1} - z^k\|^2 \\
\leq\ &\varphi'(E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) - \bar{E}) \\
&\cdot \left(E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) - E_{\bar{\alpha}}(z^{k+1}, z^k, \gamma^k)\right) \\
\leq\ &\varphi\left(E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) - \bar{E}\right) \\
&- \varphi\left(E_{\bar{\alpha}}(z^{k+1}, z^k, \gamma^k) - \bar{E}\right).
\end{aligned}$$

Combining the above two inequalities, we obtain

$$\begin{aligned}
&\frac{4}{\rho}\left(\frac{\sqrt{2}}{2}\rho\|z^{k-1} - z^{k-2}\| + (\bar{\alpha}+\rho)\|z^k - z^{k-1}\|\right) \\
&\cdot \Big[\varphi\left(E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) - \bar{E}\right) \\
&\qquad - \varphi\left(E_{\bar{\alpha}}(z^{k+1}, z^k, \gamma^k) - \bar{E}\right)\Big] \\
\geq\ &\|z^{k+1} - z^k\|^2.
\end{aligned}$$

Multiplying both sides of this inequality by 4 and taking the square root, and by $2ab \leq a^2 + b^2$, we have

$$\begin{aligned}
4\|z^{k+1} - z^k\| \leq\ &\|z^k - z^{k-1}\| + \|z^{k-1} - z^{k-2}\| \\
&+ \frac{16(\bar{\alpha}+\rho)}{\rho}\Big[\varphi\left(E_{\bar{\alpha}}(z^k, z^{k-1}, \gamma^{k-1}) - \bar{E}\right) \\
&- \varphi\left(E_{\bar{\alpha}}(z^{k+1}, z^k, \gamma^k) - \bar{E}\right)\Big].
\end{aligned}$$

Summing up the above inequality for $i = k_0+1, \ldots, k$, we have

$$\begin{aligned}
\sum_{i=k_0}^{k} 4\|z^{i+1} - z^i\| \leq\ &\sum_{i=k_0}^{k}\left(\|z^i - z^{i-1}\| + \|z^{i-1} - z^{i-2}\|\right) \\
&+ \frac{16(\bar{\alpha}+\rho)}{\rho}\Big[\varphi\left(E_{\bar{\alpha}}(z^{k_0}, z^{k_0-1}, \gamma^{k_0-1}) - \bar{E}\right) \\
&- \varphi\left(E_{\bar{\alpha}}(z^{k+1}, z^k, \gamma^k) - \bar{E}\right)\Big],
\end{aligned}$$

and since $\varphi \geq 0$, we get

$$\begin{aligned}
\sum_{i=k_0}^{k} 2\|z^{i+1} - z^i\| \leq\ &2\|z^{k_0} - z^{k_0-1}\| + \|z^{k_0-1} - z^{k_0-2}\| \\
&+ \frac{16(\bar{\alpha}+\rho)}{\rho}\varphi\left(E_{\bar{\alpha}}(z^{k_0}, z^{k_0-1}, \gamma^{k_0-1}) - \bar{E}\right).
\end{aligned}$$

Taking $k \to \infty$ in the above inequality shows that

$$\sum_{k=1}^{\infty}\|z^{k+1} - z^k\| < \infty,$$

and thus the sequence $\{z^k\}$ is a Cauchy sequence. Hence the sequence $\{z^k\}$ is convergent and we get the conclusion.

B. Detailed description of VF-iDCA in the general setting

Our value function based DC algorithm as well as its convergence analysis can be straightforwardly extended to BLP in a more general setting with LL constraints.

$$\begin{aligned}
\min_{x\in X, u\in U, \lambda\in\mathbb{R}^J_+}\quad & L(x,u) \\
\text{s.t.}\quad & x \in \operatorname*{argmin}_{x'\in X}\Big\{l(x',u) + \sum_{i=1}^{J}\lambda_i P_i(x',u) \\
& \qquad\qquad \text{s.t.}\quad g(x',u) \leq 0\Big\},
\end{aligned} \tag{31}$$

where $X \subseteq \mathbb{R}^n$, $U \subseteq \mathbb{R}^d$ are closed convex sets, $L, l : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$, $P_i : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}_+,\ i = 1, \ldots, J$, $g_i : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R},\ i = 1, \ldots, m$ are convex functions defined on an open convex set containing $X \times U$ and $g : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^m$ is defined as $g(x,u) = (g_1(x,u), \ldots, g_m(x,u))$. To ensure the convergence, we assume that function $L(x,u)$ is bounded below on an open convex set containing $X \times U$.

Decoupling the hyperparameter variable λ from the regularization term, and introducing a new variable r result in the following BLP:

$$\begin{aligned}
\min_{x\in X, u\in U, \lambda\in\mathbb{R}^J_+}\quad & L(x,u) \\
\text{s.t.}\quad & x \in \operatorname*{argmin}_{x'\in X}\Big\{l(x',u) \\
& \qquad \text{s.t.}\quad g(x',u) \leq 0, \\
& \qquad\qquad P_i(x',u) \leq r_i,\ i = 1, \ldots, J\Big\}.
\end{aligned} \tag{32}$$

Denote the solution sets of the lower level programs in problem Eq. (31) and Eq. (32) by $S_p(\lambda, u)$ and $S_c(r, u)$, respectively.

BLP Eq. (32) is then reformulated as the following single-level DC program:

$$\begin{aligned}
\min_{x\in X, u\in U, \lambda\in\mathbb{R}^J_+}\quad & L(x,u) \\
\text{s.t.}\quad & l(x,u) - v(r,u) \leq 0, g(x,u) \leq 0, \\
& P_i(x,u) \leq r_i,\ i = 1, \ldots, J,
\end{aligned} \tag{33}$$

where $v(r,u)$ is the value function of the LL problem governed by (r,u)

$$v(r,u) := \min_{x'\in X}\ \{l(x',u) \text{ s.t. } g(x',u) \le 0,\quad P_i(x',u) \le r_i,\ i=1,\dots,J\}.$$

To ensure the convergence, assume that for each r belonging to any open subset of $\mathbb{R}^J_+$, the feasible region $\mathcal{F}(r,u) := \{x \in X \text{s.t.} g(x,u) \le 0, P_i(x,u) \le r_i, i = 1,\dots,J\} \neq \emptyset$ and $l(x,u)$ is bounded on $\mathcal{F}(r,u)$. Thanks to the full convexity after hyperparameter variable decoupling, $v(r,u)$ is indeed convex and locally Lipschitz continuous around every point in set $\mathbb{R}^J_{>0} \times U$ (see, e.g., Lemma 3 in (Ye et al., 2021)).

Given a current iteration (x^k, u^k, r^k) for each k, solving the LL problem parameterized by u^k and r^k

$$\min_{x\in X}\ l(x,u^k) \text{ s.t. } g(x,u^k) \le 0,\quad P_i(x,u^k) \le r_i^k,\ i=1,\dots,J, \tag{34}$$

leads to a solution $\tilde{x}^k \in S_c(r^k,u^k)$ and a corresponding KKT multiplier $(\gamma^k,\zeta^k) \in \mathcal{M}(\tilde{x}^k,u^k,r^k)$, where $\mathcal{M}(x,u,r)$ denotes the set of multipliers of the lower level problem,

$$\mathcal{M}(x,u,r) := \Big\{\lambda \in \mathbb{R}^J_+, \mu \in \mathbb{R}^m_+ \mid 0 \in \partial_x l(x,u) + \mathcal{N}_X(x) + \sum_{i=1}^J \lambda_i \partial_x P_i(x,u) + \sum_{i=1}^m \mu_i \partial_x g_i(x,u),\ \langle \mu, g(x,u)\rangle = 0,\ \lambda_i(P_i(x,u) - r_i) = 0, i = 1,\dots,J\Big\}.$$

Select

$$\xi^k \in \partial_u l(\tilde{x}^k,u^k) + \sum_{i=1}^J \gamma_i^k \partial_u P_i(\tilde{x}^k,u^k) + \sum_{i=1}^m \zeta_i^k \partial_u g_i(\tilde{x}^k,u^k). \tag{35}$$

Then by sensitivity analysis (see, e.g., Theorem 3 in (Ye et al., 2021)), it can be easily checked that $(-\gamma^k,\xi^k) \in \partial v(r^k,u^k)$. Compute $z^{k+1} := (x^{k+1},u^{k+1},r^{k+1})$ as an approximate minimizer of the strongly convex subproblem

$$\min_{x\in X, u\in U, r\in\mathbb{R}^J_+} \quad \phi_k(x,u,r) := L(x,u) + \frac{\rho}{2}\|z - z^k\|^2 + \alpha_k \max_{\substack{j=1,\dots,J\\ i=1,\dots,m}} \{0, V_k(x,u,r), P_j(x,u) - r_j, g_i(x,u)\}, \tag{36}$$

where $z := (x,u,r)$, $z^k := (x^k,u^k,r^k)$ and

$$V_k(x,u,r) := l(x,u) - l(\tilde{x}^k,u^k) - \langle \xi^k, u - u^k\rangle + \langle \gamma^k, r - r^k\rangle.$$

Denoting $\Sigma := X \times U \times \mathbb{R}^J_+$, we may introduce an inexact condition for choosing z^{k+1}

$$\operatorname{dist}(0, \partial\phi_k(z^{k+1}) + \mathcal{N}_\Sigma(z^{k+1})) \le \frac{\sqrt{2}}{2}\rho\|z^k - z^{k-1}\|. \tag{37}$$

Using above constructions, letting

$$t^{k+1} := \max_{\substack{j=1,\dots,J\\ i=1,\dots,m}} \Big\{0, V_k(z^{k+1},u^{k+1},r^{k+1}), P_j(x^{k+1},u^{k+1}) - r_j^{k+1}, g_i(x^{k+1},u^{k+1})\Big\},$$

we are ready to present VF-iDCA in Algorithm 2.

Algorithm 2 VF-iDCA

```
Take an initial point $(x^0,u^0,r^0) \in X \times U \times \mathbb{R}^J_+$; $c_\alpha, \delta_\alpha > 0$; an initial penalty parameter $\alpha_0 > 0$; tolerance $tol > 0$.
for $k = 0, 1, \dots$ do
    Solve LL problem Eq. (34). Find $\tilde{x}^k \in S_c(r^k,u^k)$ and KKT multiplier $\gamma^k, \zeta^k$.
    Solve problem Eq. (36) up to tolerance in Eq. (37). Find an approximate solution $(x^{k+1},u^{k+1},r^{k+1})$.
    Stopping test.
    Stop if $\max\{\|z^{k+1} - z^k\|, t^{k+1}\} < tol$.
    Adaptive penalty parameter update. Set
        $\alpha_{k+1} = \begin{cases} \alpha_k + \delta_\alpha, & \text{if } \max\{\alpha_k, 1/t^{k+1}\} < \frac{c_\alpha}{\Delta^{k+1}}, \\ \alpha_k, & \text{otherwise}, \end{cases}$
    where $\Delta^{k+1} := \|z^{k+1} - z^k\|$.
end for
```

Obviously similar convergence results for the general problem as in Theorems 3.2 and 3.3 can be obtained.

C. Hyperparameter Decoupling Illustration

We illustrate how to adopt the hyperparameter decoupling technique and cast the BLPs listed in Table 1 in the formulation of Eq. (4) with fully convex LL problem.

Elastic net (Zou & Hastie, 2003) is a regularized linear regression method that combines the ridge and lasso penalties. The hyperparameter optimization problem for elastic net can be formulated as the following BLP:

$$\begin{aligned}\min_{\boldsymbol{\beta}\in\mathbb{R}^p,\lambda\in\mathbb{R}^2_+} \ & \frac{1}{2}\sum_{i\in I_{\text{val}}}|b_i-\boldsymbol{\beta}^\top\mathbf{a}_i|^2 \qquad (38)\\ \text{s.t. } \boldsymbol{\beta}\in \ & \mathop{\arg\min}_{\hat{\boldsymbol{\beta}}\in\mathbb{R}^p}\Big\{\frac{1}{2}\sum_{i\in I_{\text{tr}}}|b_i-\hat{\boldsymbol{\beta}}^\top\mathbf{a}_i|^2 \\ & \qquad + \lambda_1\|\hat{\boldsymbol{\beta}}\|_1+\frac{\lambda_2}{2}\|\hat{\boldsymbol{\beta}}\|_2^2\Big\}.\end{aligned}$$

For the implementation of VF-iDCA, we adopt the hyperparameter decoupling and thus work on the following BLP:

$$\begin{aligned}\min_{\boldsymbol{\beta}\in\mathbb{R}^p,\mathbf{r}\in\mathbb{R}^2_+} \ & \frac{1}{2}\sum_{i\in I_{\text{val}}}|b_i-\boldsymbol{\beta}^\top\mathbf{a}_i|^2 \qquad (39)\\ \text{s.t. } \boldsymbol{\beta}\in \ & \mathop{\arg\min}_{\hat{\boldsymbol{\beta}}\in\mathbb{R}^p}\Big\{\frac{1}{2}\sum_{i\in I_{\text{tr}}}|b_i-\hat{\boldsymbol{\beta}}^\top\mathbf{a}_i|^2 \\ & \qquad \text{s.t. } \|\hat{\boldsymbol{\beta}}\|_1\le r_1,\ \frac{1}{2}\|\hat{\boldsymbol{\beta}}\|_2^2\le r_2\Big\}.\end{aligned}$$

Sparse group lasso (Simon et al., 2013) is a regularized linear regression method well-suited for cases where the features have a natural grouping. The sparse group lasso penalty involves both $\|\cdot\|_2$ and $\|\cdot\|_1$ norms, and encourages sparsity both on the group level and on the within group level.

Suppose the p features are divided into M groups. Denoting the corresponding coefficients of m-th group as $\boldsymbol{\beta}^{(m)}$ and the corresponding regularization parameter as λ_m (r_m for the VF-iDCA), we can write the hyperparameter optimization problem in the following way:

$$\begin{aligned}\min_{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{\lambda}\in\mathbb{R}^{M+1}_+} \ & \frac{1}{2}\sum_{i\in I_{\text{val}}}|b_i-\boldsymbol{\beta}^\top\mathbf{a}_i|^2 \qquad (40)\\ \text{s.t. } \boldsymbol{\beta}\in \ & \mathop{\arg\min}_{\hat{\boldsymbol{\beta}}\in\mathbb{R}^p}\Big\{\frac{1}{2}\sum_{i\in I_{\text{tr}}}|b_i-\hat{\boldsymbol{\beta}}^\top\mathbf{a}_i|^2 \\ & \qquad + \sum_{m=1}^{M}\lambda_m\|\hat{\boldsymbol{\beta}}^{(m)}\|_2+\lambda_{M+1}\|\hat{\boldsymbol{\beta}}\|_1\Big\}.\end{aligned}$$

Note that in the above, we consider a variation on (Simon et al., 2013) where we use a separate regularization parameter for each of the groups, rather than a single regularization parameter for the sum of the penalties across all groups, along the lines of (Feng & Simon, 2018).

For the implementation of VF-iDCA, we adopt the hyperparameter decoupling and thus work on the following BLP:

$$\begin{aligned}\min_{\boldsymbol{\beta}\in\mathbb{R}^p,\mathbf{r}\in\mathbb{R}^{M+1}_+} \ & \frac{1}{2}\sum_{i\in I_{\text{val}}}\|b_i-\boldsymbol{\beta}^\top\mathbf{a}_i\|^2 \qquad (41)\\ \text{s.t. } \boldsymbol{\beta}\in \ & \mathop{\arg\min}_{\hat{\boldsymbol{\beta}}\in\mathbb{R}^p}\Big\{\frac{1}{2}\sum_{i\in I_{\text{tr}}}\|b_i-\hat{\boldsymbol{\beta}}^\top\mathbf{a}_i\|^2 \\ & \qquad \text{s.t. } \|\hat{\boldsymbol{\beta}}^{(m)}\|_2\le r_m,\ m=1,\dots,M,\ \|\hat{\boldsymbol{\beta}}\|_1\le r_{M+1}\Big\}.\end{aligned}$$

Low-rank matrix completion is a flexible framework for reduced-rank modelling of matrix-valued data. We consider a variation explored in Feng & Simon (2018), where we have access to additional information corresponding to each row and column. Specifically, suppose that for matrix $M\in\mathbb{R}^{n\times n}$, we observe some entries M_{ij} where $(i,j)\in\Omega$ and do not have access to the rest. We denote the row features $X\in\mathbb{R}^{n\times p}$ and column features as $Z\in\mathbb{R}^{n\times p}$. We model the matrix as the sum of a low rank effect Γ and a linear combination of the row features and the column features. Denote the coefficients of row features and column features by $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, respectively. Furthermore, suppose that there is a natural grouping of the row and column features, which partitions the corresponding coefficients as $\{\boldsymbol{\theta}^{(g)}\}_{g=1}^G$ and $\{\boldsymbol{\beta}^{(g)}\}_{g=1}^G$. So with the nuclear norm penalty on Γ and the group lasso penalty on the linear part, we have the following model:

$$\begin{aligned}&\min_{\substack{\boldsymbol{\theta}\in\mathbb{R}^p,\ \boldsymbol{\beta}\in\mathbb{R}^p,\\ \Gamma\in\mathbb{R}^{n\times n},\ \boldsymbol{\lambda}\in\mathbb{R}^{2G+1}_+}} \sum_{(i,j)\in\Omega_{\text{val}}}\frac{1}{2}|M_{ij}-\mathbf{x}_i\boldsymbol{\theta}-\mathbf{z}_j\boldsymbol{\beta}-\Gamma_{ij}|^2\\ &\text{s.t. } \boldsymbol{\theta},\ \boldsymbol{\beta},\ \Gamma\in\\ &\mathop{\arg\min}_{\substack{\hat{\boldsymbol{\theta}}\in\mathbb{R}^p,\ \hat{\boldsymbol{\beta}}\in\mathbb{R}^p,\\ \hat{\Gamma}\in\mathbb{R}^{n\times n}}}\Big\{\sum_{(i,j)\in\Omega_{\text{tr}}}\frac{1}{2}|M_{ij}-\mathbf{x}_i\hat{\boldsymbol{\theta}}-\mathbf{z}_j\hat{\boldsymbol{\beta}}-\hat{\Gamma}_{ij}|^2\\ &\qquad +\lambda_0\|\hat{\Gamma}\|_*+\sum_{g=1}^{G}\lambda_g\|\hat{\boldsymbol{\theta}}^{(g)}\|_2+\sum_{g=1}^{G}\lambda_{g+G}\|\hat{\boldsymbol{\beta}}^{(g)}\|_2\Big\},\end{aligned}$$

where $\Omega=\Omega_{\text{val}}\cup\Omega_{\text{tr}}$, $\Omega_{\text{val}}\cap\Omega_{\text{tr}}=\emptyset$.

For the implementation of VF-iDCA, we adopt the hyperparameter decoupling first and obtain the following BLP:

$$\begin{aligned}&\min_{\substack{\boldsymbol{\theta}\in\mathbb{R}^p,\ \boldsymbol{\beta}\in\mathbb{R}^p,\\ \Gamma\in\mathbb{R}^{n\times n},\ \mathrm{r}\in\mathbb{R}^{2G+1}_+}} \sum_{(i,j)\in\Omega_{\text{val}}}\frac{1}{2}|M_{ij}-\mathbf{x}_i\boldsymbol{\theta}-\mathbf{z}_j\boldsymbol{\beta}-\Gamma_{ij}|^2\\ &\text{s.t. } \boldsymbol{\theta},\boldsymbol{\beta},\Gamma\in\\ &\mathop{\arg\min}_{\substack{\hat{\boldsymbol{\theta}}\in\mathbb{R}^p,\hat{\boldsymbol{\beta}}\in\mathbb{R}^p,\\ \hat{\Gamma}\in\mathbb{R}^{n\times n}}}\Big\{\sum_{(i,j)\in\Omega_{\text{tr}}}\frac{1}{2}|M_{ij}-\mathbf{x}_i\hat{\boldsymbol{\theta}}-\mathbf{z}_j\hat{\boldsymbol{\beta}}-\hat{\Gamma}_{ij}|^2\\ &\qquad\text{s.t. } \|\hat{\Gamma}\|_*\le r_0,\ \|\hat{\boldsymbol{\theta}}^{(g)}\|_2\le r_g,\ g=1,\dots,G\\ &\qquad\qquad \|\hat{\boldsymbol{\beta}}^{(g)}\|_2\le r_{g+G},\ g=1,\dots,G\Big\}\end{aligned}$$

Support vector machines (Cortes & Vapnik, 1995) are supervised learning methods for binary classification. Support vector machines solve

$$\min_{\substack{-\bar{\mathbf{w}}\leq\mathbf{w}\leq\bar{\mathbf{w}}\\ c\in\mathbb{R}}} \left\{\frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{j\in\Omega_{\text{tr}}} \max(1-b_j(\mathbf{w}^\top\mathbf{a}_j - c), 0)\right\}.$$

We note that this is a variant of support vector machines proposed by Kunapuli 2008 which incorporates feature selection through the box constraints on $\mathbf{w}$.

Recent work (Kunapuli et al., 2008) considered the T-fold cross validation method for selecting the hyperparameters λ and $\bar{\mathbf{w}}$, and treated it as a bilevel optimization problem. A given data set Ω is randomly partitioned into T pairwise disjoint subset called the validation sets $\{\Omega^t_{\text{val}}\}_{t=1}^T$. For each validation set Ω^t_{val}, the corresponding training set is $\Omega^t_{\text{tr}} := \Omega\backslash\Omega^t_{\text{val}}$. We select the best hyperparameters via minimizing the hinge loss based on the validation set and its partition by following BLP:

$$\begin{aligned}
\min_{\lambda,\bar{\mathbf{w}},\mathbf{w}^1,\dots,\mathbf{w}^T,\mathbf{c}} \quad & \Theta(\mathbf{w}^1,\dots,\mathbf{w}^T,\mathbf{c})\\
\text{s.t.}\ \ & \lambda\geq 0,\quad \bar{\mathbf{w}}_{lb}\leq\bar{\mathbf{w}}\leq\bar{\mathbf{w}}_{ub},\\
& \text{and for } t=1,\dots,T:\\
& (\mathbf{w}^t, c^t)\in \operatorname*{argmin}_{\substack{-\bar{\mathbf{w}}\ \leq\ \mathbf{w}\ \leq\ \bar{\mathbf{w}}\\ c\ \in\ \mathbb{R}}} \left\{\frac{\lambda}{2}\|\mathbf{w}\|^2\right.\\
& \qquad\left. + \sum_{j\in\Omega^t_{trn}} \max(1-b_j(\mathbf{a}_j^T\mathbf{w}-c),0)\right\},
\end{aligned}$$

with

$$\begin{aligned}
&\Theta(\mathbf{w}^1,\dots,\mathbf{w}^T,\mathbf{c})\\
&:= \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\Omega^t_{val}|}\sum_{j\in\Omega^t_{val}} \max(1-b_j(\mathbf{a}_j^T\mathbf{w}^t - c^t),0),
\end{aligned}$$

where $|\Omega|$ denotes the number of elements in set Ω.

For the implementation of VF-iDCA, we adopt the hyperparameter decoupling first and obtain the following BLP:

$$\begin{aligned}
&\min_{r,\bar{\mathbf{w}},\mathbf{w}^1,\dots,\mathbf{w}^T,\mathbf{c}} \quad \Theta(\mathbf{w}^1,\dots,\mathbf{w}^T,\mathbf{c})\\
&\text{s.t. } r\geq 0,\quad \bar{\mathbf{w}}_{lb}\leq\bar{\mathbf{w}}\leq\bar{\mathbf{w}}_{ub},\\
&\quad (\mathbf{w}^t, c^t)\in\\
&\quad \arg\min_{\mathbf{w}\in\mathbb{R}^p, c\in\mathbb{R}} \left\{\sum_{j\in\Omega^t_{trn}} \max(1-b_j(\mathbf{a}_j^T\mathbf{w}-c),0)\right.\\
&\qquad\qquad\qquad \left. \text{s.t. } \frac{1}{2}\|\mathbf{w}\|^2\leq r, -\bar{\mathbf{w}}\leq\mathbf{w}\leq\bar{\mathbf{w}}\right\}.
\end{aligned}$$

D. Numerical experiments

D.1. Numerical experiments on synthetic data

D.1.1. Elastic Net

We simulate data in a similar manner as Feng & Simon (2018) as follows. We draw $\mathbf{a}_i\in\mathbb{R}^p$ from a $N(\mathbf{0},\boldsymbol{I})$ distribution with $\text{cor}(a_{ij}, a_{ik}) = 0.5^{|j-k|}$. We draw the response $\mathbf{b}$ according to $b_i = \boldsymbol{\beta}^\top\mathbf{a}_i + \sigma\epsilon_i$, where $\boldsymbol{\beta}$ is randomly generated such that β_i is either 0 or 1 and $\sum_{i=1}^p \beta_i = 15$; $\boldsymbol{\epsilon}$ is sampled from the standard Gaussian distribution, and σ was chosen such that the signal-to-noise ratio SNR $\triangleq \|A\boldsymbol{\beta}\|/\|\mathbf{b}-A\boldsymbol{\beta}\|$ was 2.

For grid search, random search and TPE, we defined $u_1 = \log_{10}(\lambda_1)$ and $u_2 = \log_{10}(\lambda_2)$, and searched over $u_1, u_2\in[-5,-2]$, as in (Feng & Simon, 2018). Grid search was performed on a 10×10 uniformly-spaced grid. Random search was performed with 100 uniform random samples. The space used in TPE for both u_1 and u_2 was a uniform distribution on $[-5, 2]$. For IGJO, the initial guesses for λ_1 and λ_2 were 0.01 and 0.01, respectively. For IFDM, the inital guesses for λ_1 and λ_2 were were $0.01\lambda_{\text{max}}$ as in (Bertrand et al., 2021), where $\lambda_{\text{max}} = 1/|I_{\text{tr}}|\max(A_{\text{tr}}^\top\mathbf{b}_{\text{tr}})$. The maximum number of iterations of IFDM was set to be 50. For VF-iDCA, the initial guesses for r_1 and r_2 were 10 and 5, respectively. VF-iDCA was stopped when

$$\max\left\{\frac{\|\mathbf{z}^{k+1}-\mathbf{z}^k\|}{\sqrt{1+\|\mathbf{z}^k\|^2}}, t^{k+1}\right\} < tol, \qquad (42)$$

and we set $tol = 0.1$.

D.1.2. Sparse Group Lasso

The data generation method was mainly referred to Feng & Simon (2018) as follows. Each dataset contains 100 training data, 100 validation data and 100 test data. Each $\mathbf{a}_i\in\mathbb{R}^p$ were sampled from the standard normal distribution. The response $\boldsymbol{b}$ was generated by $b_i = \boldsymbol{\beta}^\top\mathbf{a}_i + \sigma\epsilon_i$, where $\boldsymbol{\beta} = \left[\boldsymbol{\beta}^{(\mathbf{1})}, \boldsymbol{\beta}^{(\mathbf{2})}, \boldsymbol{\beta}^{(\mathbf{3})}\right]$, $\boldsymbol{\beta}^{(i)} = (1,2,3,4,5,0,\dots,0)$, for $i=1,2,3$. $\boldsymbol{\epsilon}$ are generated from the standard normal distribution, and σ was chosen such that the SNR is 2.

For grid search, we did the search on two hyperparameters μ_1, μ_2 such that $\lambda_m = 10^{\mu_1}$, for $m = 1,2,\dots,M$ and $\lambda_{M+1} = 10^{\mu_2}$, we use a 10×10 uniform-spaced grid on $[-3,1]\times[-3,1]$. For random search and TPE method, we search over the transformed variables u_m, where $u_m = \log_{10}(\lambda_m)$, for $m = 1,2,\dots,M+1$, and the space of u_m is defined as a uniform distribution on $[-3,1]$. For implicit differentiation method, we use the initial guess $[0.01, 0.01,\dots,0.01]$ for $\boldsymbol{\lambda}$. For VF-iDCA, we use the initial guess $[10,10,\dots,10]$ for $\mathbf{r}$. The stopping criterion used for VF-iDCA here was Eq. (42) with $tol = 0.05$. In all experiments, the features are grouped by order, i.e., the first p/M features are grouped as the first group, the next p/M features are grouped as the second group, etc.

D.1.3. Low-rank matrix completion

The data generation method was also referred to Feng & Simon (2018) as follows. We took two entries per row and column as the training set Ω_{tr} and one entry per row and column as the validation set Ω_{val}. The rest entries were collected as test set Ω_{test}. The row features are grouped into 12 groups of 3 covariates each, and same for the columns features, i.e., $p = 36,\ G = 12$.

The true coefficients chosen as $\boldsymbol{\alpha}^{(g)} = g\mathbf{1}_3$ for $g = 1, \ldots, 4$ and $\boldsymbol{\beta}^{(g)} = g\mathbf{1}_3$ for $g = 1,\ 2$. The rest of the coefficients were zero. We generated rank-one effect matrices $\Gamma = \mathbf{u}\mathbf{v}^\top$, where $\mathbf{u}$ and $\mathbf{v}$ were sampled from a standard normal distribution. The row features and column features X and Z were sampled from a standard normal distribution, and scaled such that the Frobenius norm of $X\boldsymbol{\alpha}\mathbf{1}^\top + (Z\boldsymbol{\beta}\mathbf{1}^\top)^\top$ was the same as Γ. The matrix data were generated by $M_{ij} = \mathbf{x}_i\boldsymbol{\alpha} + \mathbf{z}_j\boldsymbol{\beta} + \Gamma_{ij} + \sigma\epsilon_{ij}$, where ϵ_{ij} were generated from the standard normal distribution, and σ was chosen such that the SNR is 2.

For grid search, we did the search on two hyperparameters $\mu_1,\ \mu_2$ such that $\lambda_0 = 10^{\mu_1}$ and $\lambda_g = 10^{\mu_2}$, for $g = 1, \ldots, 2G$, we use a 10×10 uniform-spaced grid on $[-3.5, -1] \times [-3.5, -1]$ as Feng & Simon (2018) did. For random search and TPE method, we search over the transformed variables u_g, where $u_g = \log_{10}(\lambda_m)$, for $m = 0, 1, 2, \ldots, 2G$, and the space of u_g is defined as a uniform distribution on $[-3.5, -1]$. For IGJO, we use the initial guess $[0.005, 0.005, \ldots, 0.005]$ for $\boldsymbol{\lambda}$. For VF-iDCA, we use the initial guess $[1, 0.1, 0.1, \ldots, 0.1]$ for $\mathbf{r}$. The stopping criterion used for VF-iDCA here was Eq. (42) with $tol = 0.05$. In all experiments, the features are grouped by order, i.e., the first 3 features are grouped as the first group, the next 3 features are grouped as the second group, etc.

D.2. Application to real data

D.2.1. Elastic Net

The numerical settings for these method are kept the same as in Section 5.1.1. For grid search, random search and TPE method, we did the search on the logarithmic hyperparameters $u_1 = \log_{10}(\lambda_1)$ and $u_2 = \log_{10}(\lambda_2)$, and the range of both u_1 and u_2 is $[-5, 2]$ as in Feng & Simon (2018). Grid search was performed on a 10×10 uniform-spaced grid. Random search was performed with 100 uniformly random sampling. The space used in TPE method for both u_1 and u_2 is deifned as a uniform distribution on $[-5, 2]$. For IGJO, the initial guesses for λ_1 and λ_2 were 0.01 and 0.01, respectively. For IFDM, the inital guesses for λ_1 and λ_2 were $0.01\lambda_{\text{max}}$ as in Bertrand et al. (2021), where $\lambda_{\text{max}} = 1/|I_{\text{tr}}| \max(A_{\text{tr}}^\top \mathbf{b}_{\text{tr}})$. The maximum number of iterations of IFDM was set to be 10. For VF-iDCA, the initial guesses for r_1 and r_2 are 10, 5. VF-iDCA was stopped when Eq. (42) was satisfied with $tol = 0.1$.

D.2.2. Support Vector Machine

For grid search and random search, we did the search over two hyperparameters μ_1 and μ_2, where $\lambda = 10^{\mu_1}$, $\bar{\mathbf{w}} = (10^{\mu_2}, \ldots, 10^{\mu_2})^\top$, the range of μ_1 is -4 to 4, and the range of μ_2 is -6 to 2. For TPE method, we search the $\log_{10}(\lambda)$ in $[-4, 4]$, and $\log_{10}(\bar{w}_i)$ in $[-6, 2]$. However, since TPE is slow when the dimension is high, we set the maximum number of iteration to be 10. And we also tested TPE method on the simplified model, i.e., the same setting as the search methods, and denote such a method as TPE2. For the simplified version, we set the maximum number of iterations to be 100. For VF-iDCA, the initial guess of r is 10 and the initial guess of $\bar{\mathbf{w}}$ is $(10^{-6}, \cdots, 10^{-6})^\top$. We implement VF-iDCA with two different stopping criteria, i.e., Eq. (42) with $tol = 0.01$ and $tol = 0.1$. We denote the one with $tol = 0.01$ by VF-iDCA, and the one with $tol = 0.1$ by VF-iDCA-t. We set $\bar{\mathbf{w}}_{lb} = 10^{-6}$ and $\bar{\mathbf{w}}_{ub} = 10$ in the model.