Journal manuscript No.
(will be inserted by the editor)

Using Filter Methods to Guide Convergence for ADMM, with Applications to Nonnegative Matrix Factorization Problems

Robert J. Baraldi · Sven Leyffer ·
Stefan M. Wild

Received: / Accepted:

Abstract Nonconvex, nonlinear cost functions arise naturally in physical inverse problems and machine learning. The alternating direction method of multipliers (ADMM) has seen extensive use in these applications, despite exhibiting uncertain convergence behavior in many practical nonconvex settings, and struggling with general nonlinear constraints. In contrast, filter methods have proved effective in enforcing convergence for sequential quadratic programming methods and interior point methods with feasibility criteria. We develop an ADMM-filter method for highly nonlinear and nonconvex problems. We show convergence under mild assumptions for several types of coordinate descent schemes, and demonstrate our algorithm on nonnegative matrix factorization and completion problems in imaging and chemical spectrum analysis.

Keywords Nonconvex optimization, alternating direction method of multipliers, augmented Lagrangian, filter methods

Mathematics Subject Classification (2000) 90C26, 90C30, 49M37

Robert Baraldi
Optimization and Uncertainty Quantification,
Sandia National Laboratories, P.O. Box 5800,
Albuquerque, NM 87125, USA;
E-mail: rjbaral@sandia.gov

Sven Leyffer
Mathematics and Computer Science Division,
Argonne National Laboratory,
Lemont, IL 80439, USA;
E-mail: leyffer@anl.gov

Stefan M. Wild
Applied Mathematics and Computational Research Division,
Lawrence Berkeley National Laboratory,
Berkeley, CA 94720, USA;
E-mail: wild@lbl.gov

1 Introduction

We address the nonlinear multiblock optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) \quad \text{s.t. } \mathbf{c}(\mathbf{x}) = \mathbf{0}, \; \mathbf{x} \in \mathcal{X}, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{c} : \mathbb{R}^n \to \mathbb{R}^m$ are twice continuously differentiable, $\mathcal{X} \subseteq \mathbb{R}^n$ is a set comprising simple bounds defined by the Cartesian product $\mathcal{X} \equiv \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_p$, and $\mathbf{x}$ can be decomposed into p blocks $\mathbf{x} \coloneqq [x_1; \ldots; x_p] \in \mathbb{R}^n$ where $x_i \in \mathbb{R}^{n_i}$ and $n = \sum_{i=1}^p n_i$. The assumption that f and $\mathbf{c}$ be twice continuously differentiable is common in filter methods [50] but has not been extended to block-separable variables for that algorithm class. Because we utilize the multiblock structure in an augmented Lagrangian context, we can view the algorithm we propose as an alternating direction method of multipliers (ADMM)-like algorithm. We employ recent advances in splitting and filter methods to prove convergence for this filter-guided ADMM-like algorithm. The contribution is threefold: (1) we show filter methods can accommodate multiblock, nonconvex, nonlinear cost functions; (2) we show that ADMM methods can converge with nonlinear constraints when guided by filters; and (3) we demonstrate our ADMM-filter method on numerical examples with nonlinear constraints, which can also correct bad initial guesses for the augmented Lagrangian penalty parameter.

1.1 Background

ADMM is an extremely popular splitting method first introduced in [18] and later proposed as a method for variational problems in [26, 28]. The basic structure and elementary convergence properties for 2-block ADMM with convex f_1, f_2 and linear $\mathbf{c}$ are summarized in [10, 20] along with numerous applications of ADMM such as power system state estimation, principle component analysis, tensor completion, and robust graphical LASSO. The classical 2-block ADMM [18, 28, 40] is typically applied as

$$\min_{\mathbf{x}\coloneqq[x_1;x_2]\in\mathcal{X}} f(\mathbf{x}) \coloneqq f_1(x_1) + f_2(x_2) \quad \text{s.t.} \quad \mathbf{c}(\mathbf{x}) \coloneqq A_1x_1 + A_2x_2 - \mathbf{b} = \mathbf{0} \tag{2}$$

with linear constraints, or $\mathbf{c}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$. The feasible sets, $\mathcal{X}$ and $\{\mathbf{x} \mid A_1x_1 + A_2x_2 = \mathbf{b}\}$, are typically closed convex sets; additionally f_1, f_2 are convex, possibly nonsmooth functions with a nonempty solution set. Recent developments [27, 32, 52] have extended ADMM convergence to multiblock, nonconvex objective functions:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b} \tag{3}$$

for $A_0x_0 + \ldots + A_px_p = \mathbf{A}\mathbf{x}$. In [32, 52], additional structure on f is often assumed, such as block-decomposable regularizers $f(\mathbf{x}) = h(\mathbf{x}) + \sum_{i=1}^p g_i(x_i)$ for h smooth and g_i possibly nonsmooth or nonconvex. The basic routine, given in Algorithm 1 with iterates $\mathbf{x}^{(k)}$ (see Section 1.2 for a definition of the notation used), minimizes the augmented Lagrangian for (3) in each block with a given penalty parameter ρ:

$$\mathcal{L}_\rho(\mathbf{x},\mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x}-\mathbf{b}) + \frac{\rho}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{b}\right\|^2. \tag{4}$$

Note that our formulation (1) is more general than (3) in the admission of *nonlinear constraints* $c(\mathbf{x})$. To the best of our knowledge, ADMM with wholly nonlinear constraints does not exist in the literature.

Algorithm 1: Basic ADMM p-block splitting for (3)

```
Data: function $f$, matrix $\mathbf{A}$, vector $\mathbf{b}$, augmented Lagrangian parameter $\rho > 0$,
      and initial $\mathbf{x}^{(0)}, \mathbf{y}^{(0)}$
Result: $\mathbf{x} \leftarrow \arg\min_{\mathbf{x}} f(\mathbf{x})$ s.t. $\mathbf{A}\mathbf{x} = \mathbf{b}$ in (3)
for $k = 0, 1, \ldots$ do
    for $i = 1, \ldots, p$ do
        $x_i^{(k+1)} = \arg\min_{x_i} \mathcal{L}_\rho([\mathbf{x}_1^{(k)}; \ldots x_i^{(k)}; \ldots \mathbf{x}_p^{(k)}], \mathbf{y}^{(k)})$
    $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b})$
```

The 2-block problem is significantly different from the p-block problem for $p > 2$, even for a convex objective f. For example, ADMM diverges for any ρ when $p = 3$ blocks and $f_1, f_2, f_3 \equiv 0$ [11]. However, 3-block ADMM can converge when f_3 is strongly convex with condition number $\kappa \in [1, 1.0798)$ [39]. Davis and Yin [15] create a three-operator splitting algorithm that converges without strong convexity assumptions. Such operator-splitting perspectives for ADMM and its variations are discussed in [19] as well as the original ADMM texts [18]. For $p > 3$, when $f_1, \ldots, f_p$ are all strongly convex and $\rho > 0$ is sufficiently small, then multiblock ADMM is convergent [31]. Similar results can be seen in [35–38]. Multiblock ADMM convergence has been demonstrated through variants such as proximal ADMM [9], linearized ADMM [12, 16, 47, 56], and proximal Jacobi ADMM [16, 48, 51].

Convergence for nonconvex and possibly nonsmooth $f(\mathbf{x})$ with structure $h(\mathbf{x}) + \sum_{i=0}^{p} g_i(x_i)$ is shown in [52], with applications in optimization and machine learning [57, 58]. The objective assumptions are $f(\mathbf{x})$ prox-regular; $h(\mathbf{x})$ Lipschitz differentiable; and either (1) g_0 is lower semi-continuous and $g_i(x_i)$ is restricted prox-regular for $i = 1, \ldots, p$ or (2) bounded subdifferentials $\mathbf{d} \in \partial g_0(x_0)$ and $g_i(x_i)$ are continuous and piecewise linear. Additional assumptions include coercivity of the objective over the feasible set, prox-regularity [45], constraints on the image of A_i for $i = 1, \ldots, p$, and existence of Lipschitz subminimization paths. Under these assumptions, Wang et al. [52] show that Algorithm 1 with p blocks converges for sufficiently large ρ (with lower bound based on restricted prox-regularity parameter) for any starting point $(\mathbf{x}, \mathbf{y})$, and generates a bounded sequence with at least one limit point that is a stationary point for the augmented Lagrangian (4). If (4) is a Kurdyka-Łojasiewicz function (see [2]), then it converges globally to the unique limit point.

More recently, the authors in [32] consider (1) with regularized structure

$$\min_{\mathbf{x},z} f(\mathbf{x}) + \sum_{i=1}^{p} g_i(x_i) + h(z) \quad \text{s.t.} \quad \sum_{i=1}^{p} A_i x_i + \mathbf{B}\mathbf{z} = \mathbf{b},$$

where h is differentiable, f is a nonconvex nonsmooth function, and g_i are proper lower semi-continuous functions. The Kurdyka–Łojasiewicz property is used to prove global convergence of their ADMM routine with inertial updates on the primal variables. When no inertial terms are used, the algorithm reduces to ADMM that employs minimization-majorization principles in each block update. In contrast to our work, [32] and [52] have more lax assumptions (which may be unverifiable *a priori*) on f but still require $\mathbf{c}$ to be linear. The closest ADMM routine for nonlinear $\mathbf{c}$ is presented in [27], which allows for the constraints to be multiaffine under a large set of assumptions on f, similar to [52]. An example of multiaffine ADMM comes from [30], which extends the nonconvex formulation to include bilinearly constrained optimization problems in the form of nonnegative matrix factorization (NMF). We presently generalize this to fully nonlinear $\mathbf{c}(\mathbf{x})$.

Filter methods provide an alternative to penalty methods as a way to solve nonconvex optimization problems [23]. First proposed by [21] and generalized to sequential quadratic programming methods in [22], filter methods view the objective and constraint violation as a biobjective optimization problem that jointly minimizes f and $\|\mathbf{c}\|$. More recently, filter methods have been employed to force convergence of augmented Lagrangian methods for nonlinear optimization [50]; this work's algorithm restores feasibility when necessary and does not require forcing sequences for first-order error for the same conditions as our setting (1). The filter provides a mechanism to enforce feasibility via restoration and penalty parameter increases similar to [25], where a minimum $\rho_{\min} > 0$ is established that depends on the eigenvalues of the Hessian of the objective and the singular values of the Jacobian. We utilize the filter's feasibility enforcement to extend ADMM's problem class to nonlinear constraints. To the best of our knowledge, the multiblock structure of our routine has not been extended to this analysis.

1.2 Notation

We denote $\mathbb{R}$ as the real number set, $\mathbb{R}_+$ as the positive real numbers; other sets are represented by calligraphic letters, such as $\mathcal{X}$. The cardinality of sets is represented by $|\cdot|$. We use $\|\cdot\|$ to denote the Euclidean norm. The symbol ρ is the penalty parameter; p is the number of blocks. The letter i is typically used to represent blocks, whereas j, k represent filter/inner iteration and optimal/outer iteration indices, respectively. We use bold face (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{A}$) to represent block vectors and matrices, unbolded type for individual blocks (e.g., $\mathbf{x} = [x_1; \ldots; x_p]$). We let $\mathbf{x}_{<i} := [x_1; \ldots; x_{i-1}] \in \mathbb{R}^{n_1+n_1+\ldots+n_{i-1}}$ and $\mathbf{x}_{>i} := [x_{i+1}; \ldots; x_p] \in \mathbb{R}^{n_{i+1}+\ldots+n_p}$ with $\mathbf{x}_{<1}$ and $\mathbf{x}_{>p}$ being null variables. If

$\mathcal{A} \subseteq \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^n$, then $\text{dist}(\mathbf{x}; \mathcal{A}) = \inf\{\|\mathbf{a} - \mathbf{x}\| | \mathbf{a} \in \mathcal{A}\}$ is the Euclidean distance from $\mathbf{x}$ to $\mathcal{A}$. If $\mathcal{A}$ is closed and convex, $\text{proj}_{\mathcal{A}}(\mathbf{x})$ denotes the unique projection of $\mathbf{x}$ onto $\mathcal{A}$. The symbol $\mathcal{F}$ denotes with filter, while $\mathcal{L}_\rho$ signifies the augmented Lagrangian with parameter ρ. The gradient of a function is always taken with respect to $\mathbf{x}$ and is denoted $\nabla\mathcal{L}$. The gradient of a particular block is given by $\nabla_i\mathcal{L} \in \mathbb{R}^{n_i}$. The Greek letters η and ω represent the feasibility of the nonlinear program and the first-order error, respectively. Slightly modified notation for the block coordinate descent section is described separately in Section 4.

1.3 Roadmap

In Section 2, we introduce filters as an algorithm method and state our algorithm. Section 3 states the filter convergence as in [50], leaving out sufficient decrease. Section 4 proves the aforementioned sufficient decrease of block updates needed by the filter. Section 5 demonstrates our algorithm on two test cases: nonnegative matrix factorization and its completion variant. Section 6 demonstrates a highly nonlinear example where we reconstruct chemical spectra intensities.

2 Filters for Augmented Lagrangians

Recall that we wish to solve (1) via the Lagrangian and augmented Lagrangian

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) - \mathbf{y}^T\mathbf{c}(\mathbf{x}), \quad \mathcal{L}_\rho(\mathbf{x}, \mathbf{y}) = \mathcal{L}(\mathbf{x}, \mathbf{y}) + \tfrac{\rho}{2}\|\mathbf{c}(\mathbf{x})\|^2. \tag{5}$$

where $\mathbf{y} \in \mathbb{R}^m$ is a vector of the Lagrange multipliers associated with $\mathbf{c}(\mathbf{x}) = \mathbf{0}$. The augmented Lagrangian more closely enforces feasibility by penalizing the constraint violation scaled by some parameter $\rho > 0$. If we designate iterations via k for penalty parameter $\rho^{(k)}$ and multipliers $\mathbf{y}^{(k)}$, minimizing $\mathcal{L}_{\rho^{(k)}}(\mathbf{x}, \mathbf{y}^{(k)})$ over $\mathbf{x} \in \mathcal{X}$ yields the simply constrained subproblem

$$\min_{\mathbf{x} \in \mathcal{X}} \mathcal{L}_{\rho^{(k)}}(\mathbf{x}, \mathbf{y}^{(k)}), \tag{6}$$

whose solution may not be unique. A basic augmented Lagrangian scheme takes steps in $\mathbf{x}$ and $\mathbf{y}$, approximately solving (6) with bounds $\mathcal{X}$ to obtain $\mathbf{x}^{(k+1)}$, and then either updating the multipliers $\mathbf{y}$ via the first-order update

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \rho^{(k)}\mathbf{c}(\mathbf{x}^{(k+1)}) \tag{7}$$

or keeping $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)}$ and increasing $\rho^{(k)}$ [25]. Augmented Lagrangian algorithms have experienced renewed interest in recent years as they have demonstrated desirable scalability properties [13, 14]. They also enjoy satisfactory convergence theory when $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ are simple bounds, that is, where $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$ [50]. The feasible set in (1), $\mathcal{X} \subseteq \mathbb{R}^n$, is assumed to admit a well-defined projection since $\mathcal{X}$ is nonempty, closed, and convex. Simple bound constraints $\mathcal{X} = \{\mathbf{x} : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ have the projection

$$\left[\underset{\mathcal{X}}{\operatorname{proj}}(\mathbf{x})\right]_i = \min\{\max\{l_i, x_i\}, u_i\}, \qquad i = 1, \dots, n. \tag{8}$$

Simple restriction operators also fit within this framework. For $\mathcal{S} \subset \{1, 2, \dots, n\}$ be the set of observed entries, then a sampling operator $\mathcal{A}_{\mathcal{S}} : \mathbb{R}^n \to \mathbb{R}^n$ with elementwise action admits

$$\mathcal{A}_{\mathcal{S}}(\mathbf{x}) = \underset{\mathcal{X}_{\mathcal{S}}}{\operatorname{proj}}(\mathbf{x}) = \begin{cases} x_i, & i \in \mathcal{S}, \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\mathcal{X}_{\mathcal{S}}$ is defined by:

$$\mathcal{X}_{\mathcal{S}} = \{x_i : l_i \leq x_i \leq u_i\}_{i=1,\dots,n}, \quad \text{with} \quad \begin{cases} l_i = -\infty, u_i = \infty, & i \in \mathcal{S}, \\ l_i = u_i = 0, & i \notin \mathcal{S}. \end{cases}$$

This relationship is utilized in Section 5 when we employ completion on the set of observed image entries.

2.1 Optimality, Feasibility, and Filters

The first-order optimality and feasibility conditions of (1) for a local minimum $\mathbf{x}^*$ can be expressed by

$$\underset{\mathcal{X}}{\operatorname{proj}}\left(\mathbf{x}^* - \nabla\mathcal{L}_\rho(\mathbf{x}^*, \mathbf{y})\right) = \mathbf{x}^*, \tag{10a}$$

$$\mathbf{c}(\mathbf{x}^*) = \mathbf{0}, \tag{10b}$$

where $\nabla\mathcal{L}_\rho(\mathbf{x}, \mathbf{y})$ is the gradient with respect to $\mathbf{x}$. Let the sequences $\eta^{(k)}$ and $\omega_\rho^{(k)}$ be measures of optimality and feasibility from (10),

$$\eta(\mathbf{x}) := \|\mathbf{c}(\mathbf{x})\|, \tag{11a}$$

$$\omega_\rho(\mathbf{x}, \mathbf{y}) := \left\|\underset{\mathcal{X}}{\operatorname{proj}}\left(\mathbf{x} - \nabla\mathcal{L}_\rho(\mathbf{x}, \mathbf{y})\right) - \mathbf{x}\right\|. \tag{11b}$$

We note that $\omega_0(\mathbf{x}, \mathbf{y})$ is the dual feasibility error of the problem (1). We use ω_0 because (7) implies that $\nabla\mathcal{L}(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}) = \nabla\mathcal{L}_{\rho^{(k)}}(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k)})$ and hence from [50] we observe that $\omega_0(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}) = \omega_{\rho^{(k)}}(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k)})$, which is the dual feasibility error of (6). The augmented Lagrangian filter, denoted $\mathcal{F}$, monitors the dual infeasibility error of the original problem while solving (6).

Definition 2.1 (Augmented Lagrangian Filter and Acceptance) A filter $\mathcal{F}$ is a list of pairs $(\eta^{(l)}, \omega^{(l)}) := (\eta(\mathbf{x}^{(l)}), \omega_0(\mathbf{x}^{(l)}, \mathbf{y}^{(l)}))$ such that no pair dominates another pair; i.e., there exist no pairs $(\eta^{(l)}, \omega^{(l)})$, $(\eta^{(k)}, \omega^{(k)})$, $l \neq k$ such that $\eta^{(l)} \leq \eta^{(k)}$ and $\omega^{(l)} \leq \omega^{(k)}$. A point $(\mathbf{x}, \mathbf{y})$ is acceptable to the filter $\mathcal{F}$ iff for $0 < \gamma, \beta < 1$,

$$\eta(\mathbf{x}) \leq \beta\eta^{(l)} \quad \text{or} \quad \omega_0(\mathbf{x}, \mathbf{y}) \leq \omega^{(l)} - \gamma\eta(\mathbf{x}), \ \forall\ (\eta^{(l)}, \omega^{(l)}) \in \mathcal{F}. \tag{12}$$

The mechanism of the filter forces new updates to either (i) push $\eta(\mathbf{x}) \to 0$ or (ii) push $\omega_0(\mathbf{x}, \mathbf{y}) \to 0$ and $\eta(\mathbf{x}) \to 0$, guaranteeing a feasible if not also first-order optimal solution. Per the conclusions in [50], the filter envelope parameters ensure both that iterates do not accumulate at points where $\eta > 0$. Additionally, the multipliers need not remain bounded and our convergence proof assumes that there are no feasible points at infinity.

2.2 ADMM-Filter Algorithm

We extend [50, Algorithm 2] to block descent primal updates, which yields an ADMM-like scheme. Within this algorithm, we use two primary iteration notations: k for outer/optimal iterations and j for inner/filter iterations. The outer (k-indexed) iterations are used to test first-order optimality, feasibility, and overall convergence denoted by $\mathbf{x}^{(k)}$. The inner (j-indexed) iterations taken within the filter until we reach filter acceptability as defined in Definition 2.1 are denoted by $\mathbf{x}^{(j)}$. Our ADMM-Filter method, formally described in Algorithm 2, can apply multiple forms of block descent. Line 6 of Algorithm 2 chooses a descent direction that obtains sufficient decrease as defined by

$$\Delta\mathcal{L}^{(j)}_{\rho^{(k)}} := \mathcal{L}_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)}) - \mathcal{L}_{\rho^{(k)}}(\mathbf{x}^{(j+1)}, \mathbf{y}^{(k)}) \geq \sigma\omega_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)}) \qquad (13)$$

for $\sigma \in (0, 1]$. We detail ways of achieving this decrease in Section 4. Briefly, (13) can be satisfied by selecting the single block projected gradient step from $\mathbf{x}^{(j)}$ which yields the largest predicted decrease of the augmented Lagrangian. Another method is to simply run one cycle of projected gradient decent in each block, or, as in ADMM, minimize in each block. Unless we are stationary in every block, such a descent direction will exist for at least one block. In Sections 5 and 6, we utilize cyclic block projected gradient descent for early filter iterations and cyclic minimization for later iterations; empirically, we find this improves performance as similar progress can be made earlier in the filter iteration at a cheaper cost. The filter step is used to gauge whether the decrease from (13) yielded by the primal block-coordinate descent step successfully makes progress toward first-order optimality and feasibility. If not, we check restoration conditions [50, Eqs. (3.15) and (3.16)]:

$$\eta(\mathbf{x}) = \eta(\mathbf{x}^{(j+1)}) \geq \beta U, \qquad (14a)$$

$$\omega_{\rho^{(k)}}(\mathbf{x}^{(j+1)}, \mathbf{y}^{(k)}) \leq \epsilon \quad \text{and} \quad \eta(\mathbf{x}^{(j+1)}) \geq \beta\eta_{\min}, \qquad (14b)$$

which, if satisfied, trigger a restoration phase that either: (i) finds a feasible point and increases the penalty parameter, or (ii) minimizes the constraints and terminates. We use the penalty parameter update scheme

$$\zeta = \max\left(1.1, (\eta^{(j)})^2/\Delta\mathcal{L}^{(j)}_{\rho^{(k)}}\right), \qquad (15)$$

where $(\eta^{(j)})^2/\Delta\mathcal{L}^{(j)}_{\rho^{(k)}}$ is motivated by [50, Lemma 6 and Equation 4.2]; it is and aggressive scaling which exhibits the best performance when $\rho^{(k)}$ is poor, i.e. too small. We stop when the first-order error (11b) and feasibility (11a) are less than some $\epsilon \in (0,1)$

$$\omega_0(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) < \epsilon \quad \text{and} \quad \eta(\mathbf{x}^{(k)}) < \epsilon. \tag{16}$$

Algorithm 2: ADMM-F for (1)

```
Data: Functions $f$ and $\mathbf{c}$, $(\mathbf{x}^{(0)},\ \mathbf{y}^{(0)})$, $\rho^{(0)}$, and $k \leftarrow 0$.
while $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ not optimal according to (16) do
    $j \leftarrow 0$, Restflag $\leftarrow 0$;
    Declare temporary variable: $\mathbf{x}^{(j)} \leftarrow \mathbf{x}^{(k)}$;
    while $(\eta^{(j)}, \omega^{(j)})$ not acceptable to $\mathcal{F}_k$ do
        if $j = 0$ then
            Find $\mathbf{x}^{(j+1)}$ to satisfy (13) // sufficient-decrease step
        else
            for $i = 1, \ldots, p$ // min each of $p$ blocks
             do
                $x_i^{(j+1)} \leftarrow \arg\min_{x_i \in \mathcal{X}_i} \mathcal{L}_{\rho^{(k)}}([\mathbf{x}_{<i}^{(j+1)}; x_i; \mathbf{x}_{>i}^{(j)}], \mathbf{y}^{(k)})$;
        $j \leftarrow j + 1$;
    if Restoration condition (14) holds then
        Restflag $\leftarrow 1$;
        Find $\mathbf{x}^{(j+1)}$ s.t. $(\eta^{(j+1)}, \omega^{(j+1)}) \in \mathcal{F}_k$, // exits Filter
        or $\mathbf{x}^{(j+1)} \leftarrow \arg\min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{c}(\mathbf{x})\|^2$ // terminates algorithm
        $j \leftarrow j + 1$;
    else
        Compute $\omega_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)}), \eta(\mathbf{x}^{(j)}, \mathbf{y}^{(k)})$;
    Update: $(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}) \leftarrow (\mathbf{x}^{(j)}, \mathbf{y}^{(k)} - \rho^{(k)}\mathbf{c}(\mathbf{x}^{(j)}))$;
    if $\eta^{(k)} > 0$ then
        Update: $(\eta^{(k)}, \omega^{(k)}) \leftarrow (\omega_0(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}), \eta(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}))$;
        $\mathcal{F}_{k+1} \leftarrow \{(\eta^{(k)}, \omega^{(k)})\} \cup \mathcal{F}_k$, ensuring $\eta_\ell > 0\ \forall\ \ell \in \mathcal{F}_{k+1}$;
    if Restflag $= 1$ then
        Increase Penalty $\rho^{(k+1)} \leftarrow \zeta\rho^{(k)}$   for $\zeta$ defined in (15);
    $k \leftarrow k + 1$;
```

3 Global Convergence Proof

We make the following assumption from [50].

Problem Assumption 3.1 (Differentiability and Set Compactness) *Assume that f and $\mathbf{c}$ in (1) are twice continuously differentiable, and the constraint norm satisfies $\|\mathbf{c}(\mathbf{x})\| \to \infty$ as $\|\mathbf{x}\| \to \infty$.*

Problem Assumption 3.1 implies that $f, \mathbf{c}$, and their derivatives are bounded for all iterates, and that our iterates remain in a compact set, which can be replaced by optimizing over finite bounds $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$. Algorithm 2 has three distinct outcomes (see [50]):

1. there exists an infinite sequence of restoration phase iterates $\mathbf{x}^{(k_\ell)}$, indexed by $\mathcal{R} := \{k_1, k_2, \ldots\}$, whose limit point $\mathbf{x}^* := \lim_{\ell \to \infty} \mathbf{x}^{(k_\ell)}$ minimizes the constraint violation, satisfying $\eta(\mathbf{x}^*) > 0$;
2. there exists an infinite sequence of successful major iterates $\mathbf{x}^{(k_l)}$, indexed by $\mathcal{S} := \{k_1, k_2, \ldots\}$, and the linear independence constraint qualification (LICQ) fails to hold at the limit $\mathbf{x}^* := \lim_{\ell \to \infty} \mathbf{x}^{(k_\ell)}$, which is a Fritz-John (FJ) point of (1);
3. there exists an infinite sequence of successful major iterates $\mathbf{x}^{(k_l)}$, indexed by $\mathcal{S} := \{k_1, k_2, \ldots\}$, and LICQ holds at the limit $\mathbf{x}^* := \lim_{\ell \to \infty} \mathbf{x}^{(k_\ell)}$, which is a Karush-Kuhn-Tucker (KKT) point of (1).

The entire proof for filter convergence for the setting (1) is given in [50], which we briefly summarize here. We have from [50, Lemma 2 and 3] that $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(j)}$ remain in a compact set, and there exists a neighborhood around $(\eta, \omega) = (0, 0)$ that does not contain any filter entries. [50, Lemma 4] proves that filter iterations are finite, while [50, Lemma 5 and Lemma 6] imply that $\eta_k \to 0$ and $\omega_k \to 0$ for an infinite number of outer/optimal iterations with $\rho^{(k)} < \infty$. The result in [50, Lemma 7] proves that when $\rho^{(k)} \to \infty$, any limit point $\mathbf{x}^{(k)} \to x^*$ is a FJ point, and a KKT point if the LICQ holds. It may be possible to extend the setting of (1) to $f, \mathbf{c}$ Lipschitz continuous by showing [24, Theorem 2.5] can work for such function settings; we leave this for future work. Our context is the same as [50] but with a block coordinate sufficient decrease condition instead of a single step [50, Lemma 6]. Therefore, it remains to prove that block coordinate descent steps satisfy (13); this is shown next in Section 4. With that result in hand, we cite below the proof of convergence for Algorithm 2.

Theorem 3.1 ([50, Theorem 1]) *Under Problem Assumption 3.1, either (i) Algorithm 2 terminates after a finite number of iterations at a KKT point, (i.e., for some finite k, $\mathbf{x}^{(k)}$ is a first-order stationary point with $\eta(\mathbf{x}^{(k)}) = 0$ and $\omega_0(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) = 0$), or (ii) there exists an infinite sequence of iterates $\mathbf{x}^{(k)}$ and any limit point $\mathbf{x}^{(k)} \to \mathbf{x}^*$ that satisfy one of the following:*

1. *The penalty parameter is updated finitely often, and $\mathbf{x}^*$ is a KKT point.*
2. *There exists an infinite sequence of restoration steps at which the penalty parameter is updated. If $\mathbf{x}^*$ satisfies the LICQ, it is a KKT point; otherwise, it is an FJ point.*
3. *The restoration phase converges to a minimum of the constraint violation.*

4 Sufficient Decrease with Block Coordinate Descent

Our goal now is to explore efficient coordinate descent update rules that satisfy sufficient decrease (13) for Line 6 of Algorithm 2. Coordinate-descent [17, 42, 43] and block coordinate descent [8, 29, 41, 42, 44] have a rich body of literature, with many recent advancements made in nonconvex variants. We utilize basic concepts from [5, 7, 42] to prove that we attain (13).

To avoid difficult subscript notation in our proof, we make several concessions by following the style of [5]. We shorten $\mathcal{L}(\mathbf{x}^{(j)}) \coloneqq \mathcal{L}_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)})$ as $\mathbf{y}$ and ρ are not being altered. We also define the selection matrices $\boldsymbol{E}_i \in \mathbb{R}^{n \times n_i}$, $i = 1, \ldots, p$ for which $[\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_p] = \boldsymbol{I}_n$. Our block notation can then also be written as $x_i^{(j)} = \boldsymbol{E}_i^T \mathbf{x}^{(j)}$ for all i, and $\mathbf{x}^{(j)} = \sum_{i=1}^p \boldsymbol{E}_i x_i^{(j)}$. The partial block derivatives $\nabla_i \mathcal{L}(\mathbf{x}) = \boldsymbol{E}_i^T \nabla \mathcal{L}(\mathbf{x}) \in \mathbb{R}^{n_i}$ denote the gradient of $\mathcal{L}$ with respect to the x_i. The gradient $\nabla_i \mathcal{L}(\mathbf{x}^{(j)})$ takes in the whole vector because $\mathcal{L}(\cdot)$ can be nonlinear, but the block-selection and projection onto $\mathcal{X}_i$ is separable. We establish sufficient decrease in multiple ways: (1) via taking the "maximal" projected-gradient (PG) step that yields the largest decrease on the Lagrangian, (2) cyclically taking PG steps through all the blocks, and (3) minimizing in all the blocks.

4.1 Maximally Projected Gradient Descent Direction

We use the concept of sufficient decrease with PG descent as in [42, Section 9.3]. Throughout this subsection, we define the primal variable updated in the ith block to be $\mathbf{x}_i^{(j+1)} = \left[\mathbf{x}_{<i}^{(j)}, x_i^{(j+1)}, \mathbf{x}_{>i}^{(j)}\right]$. We also need to enunciate some basic aspects of the block-separable functions and projection operators.

Definition 4.1 (Block-Lipschitz Continuity: [5, Lemma 3.2]) Suppose that $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function over $\mathbb{R}^n$ that is block-Lipschitz in the ith component:

$$\left\|\nabla_i \mathcal{L}(\mathbf{x}^{(j)}) - \nabla_i \mathcal{L}(\mathbf{x}_i^{(j+1)})\right\| \leq L_i \left\|x_i^{(j+1)} - x^{(j)}\right\|$$

for block-Lipschitz constant $L_i > 0$. Because $\mathcal{L}(\cdot)$ is continuously differentiable, we have the global upper bound for blocks that differ only up to the i-component:

$$\mathcal{L}(\mathbf{x}_i^{(j+1)}) \leq \mathcal{L}(\mathbf{x}^{(j)}) + \nabla_i \mathcal{L}(\mathbf{x}^{(j)})^T (x_i^{(j+1)} - x^{(j)}) + \frac{L_i}{2} \left\|x_i^{(j+1)} - x^{(j)}\right\|^2. \quad (17)$$

Since f and $\mathbf{c}$ are twice continuously differentiable, $\mathcal{L}(\cdot)$ is also twice continuously differentiable. Note that (17) can easily be extended to show that $\mathcal{L}(\cdot)$ is Lipschitz continuous with constant $L = \sum_{i=1}^p L_i$. We now define the PG step taken via backtracking Armijo line search along the projection arc

$$x_i^{(j+1)} \leftarrow \operatorname*{proj}_{\mathcal{X}_i} \left(x_i^{(j)} - \alpha_i \nabla_i \mathcal{L}(\mathbf{x}^{(j)})\right) \quad (18a)$$

and the resulting direction in component x_i

$$d_i(\mathbf{x}^{(j)}, \alpha_i) \coloneqq \operatorname*{proj}_{\mathcal{X}_i} \left(x_i^{(j)} - \alpha_i \nabla_i \mathcal{L}(\mathbf{x}^{(j)})\right) - x_i^{(j)} = x_i^{(j+1)} - x_i^{(j)} \quad (18b)$$

to be the block update and block step (with $d_i \in \mathbb{R}^{n_i}$). Note that these directions are all taken from the initial point $\mathbf{x}^{(j)}$. [7, Proposition 3.3.3] guarantees that $\forall\, \mathbf{x} \in \mathcal{X}$, $\exists\, \alpha_i^{\max} > 0$ such that (20) is satisfied $\forall \alpha_i \in [0, \alpha_i^{\max}]$. In practice, we do not compute $\alpha_i^{\max}$; we instead start with $\alpha_i = 1$ and backtrack. The ith block vector updates are

$$\mathbf{x}_i^{(j+1)} := \mathbf{x}^{(j)} + \boldsymbol{E}_i d_i(\mathbf{x}^{(j)}, \alpha_i). \tag{19}$$

A block is “stationary” when $d_i(\mathbf{x}^{(j)}, \alpha_i) = 0$, and the backtracking line search satisfies

$$\mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}_i^{(j+1)}) \geq -\sigma \nabla_i \mathcal{L}(\mathbf{x}^{(j)})^T d_i(\mathbf{x}^{(j)}, \alpha_i) \tag{20}$$

for some $\sigma \in (0, 1)$ [3, 7]. Monotonicity of the projection operator [3, Theorem 3.14] allows us to show

$$\left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\|^2 \leq -\alpha_i \nabla_i \mathcal{L}(\mathbf{x}^{(j)})^T d_i(\mathbf{x}^{(j)}, \alpha_i), \tag{21}$$

which we can use in conjunction with (20) to get the lower bound

$$\mathcal{L}(\mathbf{x}_i^{(j)}) - \mathcal{L}(\mathbf{x}_i^{(j+1)}) \geq \frac{\sigma}{\alpha_i} \left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\|^2. \tag{22}$$

To ensure these updates are bounded, we need to show $\alpha_i > 0$.

Proposition 4.1 (Projected Gradient Stepsize Bound) *For nonstationary blocks, the stepsize α_i produced by the Armijo line search step given by* (20) *is bounded away from zero.*

Proof Since $\mathcal{L}(\cdot)$ is Lipschitz continuous, then $\forall\, \alpha_i$ (17) with (21) yields

$$\mathcal{L}(\mathbf{x}_i^{(j+1)}) - \mathcal{L}(\mathbf{x}^{(j)}) \leq \left(\frac{L_i}{2} - \frac{1}{\alpha_i} \right) \left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\|^2.$$

Subtracting the negation of (22) from the above establishes

$$0 \leq \left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\|^2 \left(\frac{L_i}{2} - \frac{1}{\alpha_i} + \frac{\sigma}{\alpha_i} \right) \quad \text{or} \quad 0 < \frac{2(1-\sigma)}{L_i} \leq \alpha_i,$$

as $L_i > 0$ by definition and we choose $\sigma \in (0, 1)$. □

Finally, we have the following lemma from [4, Theorem 10.9] that

$$\begin{aligned} \alpha_2^{-1} \left\| d_i(\mathbf{x}^{(j)}, \alpha_2) \right\| &\geq \alpha_1^{-1} \left\| d_i(\mathbf{x}^{(j)}, \alpha_1) \right\|, \\ \left\| d_i(\mathbf{x}^{(j)}, \alpha_2) \right\| &\leq \left\| d_i(\mathbf{x}^{(j)}, \alpha_1) \right\| \end{aligned} \tag{23}$$

for $\alpha_1 \geq \alpha_2$. To prove sufficient decrease (13), we first define the set of directions yielded by block PG. Let the set $\mathcal{E}$ consist of the generalized gradient (or scaled PG step)

$$\mathcal{E} = \left\{ i = 1, \ldots, p \,|\, \alpha_i^{-1} \boldsymbol{E}_i d_i(\mathbf{x}^{(j)}, \alpha_i) \right\}.$$

Unless we are stationary, at least one direction $\xi \in \mathcal{E}$ will yield

$$\cos\theta = \frac{-\sigma \nabla \mathcal{L}(\mathbf{x}^{(j)})^T \xi}{\left\|\nabla \mathcal{L}(\mathbf{x}^{(j)})\right\| \|\xi\|} \geq \frac{-\sigma \nabla \mathcal{L}(\mathbf{x}^{(j)})^T \xi}{L \sum_{i=1}^{p} \|\xi_i\|} > \delta > 0. \tag{24}$$

We therefore select $\xi \in \mathcal{E}$ as in [42, Section 9.3]

$$\kappa(\mathcal{E}) \coloneqq \min_{v \in \mathbb{R}^n} \max_{\xi \in \mathcal{E}} \frac{v^T \xi}{\|v\| \|\xi\|} \geq \delta. \tag{25}$$

However, $v \equiv \nabla \mathcal{L}(\mathbf{x}^{(j)})$ is fixed in our case; we are simply maximizing over the directions $\xi \in \mathcal{E}$. We now show that picking the block PG step that produces the largest decrease satisfies sufficient decrease criteria (13).

Lemma 4.1 *[Largest Block PG Step Satisfies Sufficient Decrease] Let* $\{\mathbf{x}^{(j)}\}_{j \geq 0}$ *be the sequence generated by taking the projected gradient step* (18) *and direction* $\xi \in \mathcal{E}$ *that satisfies* (25) *such that* $\mathbf{x}^{(j+1)} \equiv \mathbf{x}_i^{(j+1)} = \mathbf{x}^{(j)} + \alpha_i \xi$ *from* (19). *Then for every* $j = 0, 1, 2, \ldots,$

$$\mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}^{(j+1)}) \geq \tilde{\sigma} \omega_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)}) \tag{26}$$

for $\tilde{\sigma} \in (0, 1)$.

Proof From the Armijo descent criteria (20) and our angle criteria (24),

$$\begin{aligned}
\mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}^{(j+1)}) &\geq -\sigma \nabla \mathcal{L}(\mathbf{x}^{(j)})^T \boldsymbol{E}_i d_i(\mathbf{x}^{(j)}, \alpha_i) \\
&\geq \sigma \alpha_i \delta L \left(\sum_{\ell=1}^{p} \alpha_\ell^{-1} \left\| d_\ell(\mathbf{x}^{(j)}, \alpha_\ell) \right\| \right) \\
&\geq \sigma \alpha_i \delta L \left(\sum_{\ell=1}^{p} \left\| \operatorname*{proj}_{\mathcal{X}_\ell}(x_\ell^{(j)} - \nabla_\ell \mathcal{L}(\mathbf{x}^{(j)})) - x_\ell^{(j)} \right\| \right) \\
&\geq \sigma \alpha_i \delta L \omega_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)})
\end{aligned}$$

where we used our lower bound on $\cos\theta$ and $\|\xi\|$ for $\xi \in \mathcal{E}$, and (23) respectively. Since $\sigma \delta L \alpha_i \geq 2\delta\sigma(1 - \sigma) = \tilde{\sigma}$ from Proposition 4.1, we have that $\tilde{\sigma} \in (0, .5)$ as $\delta \leq 1$. □

Therefore, simply taking the block projected gradient step that maximizes (24) on our augmented Lagrangian will yield sufficient decrease needed for [50, Lemma 6] and Theorem 3.1. Note that we do not require knowledge of the constant in front of $\omega_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)})$; this proof shows that such a strictly positive constant exists based on the Armijo line search criteria and Lipschitz constants of $\mathcal{L}(\cdot)$.

4.2 Cyclic Projected Gradient Descent

We can use similar methodology to show that one cycle of unique block coordinate PG descent yields sufficient decrease. First, we need to update our notation to reflect the changing nature of $\mathbf{x}^{(j)}$; define the primal variable updated *up to* the ith block as $\mathbf{x}_i^{(j)} = \left[\mathbf{x}_{<i}^{(j+1)}, x_i^{(j)}, \mathbf{x}_{>i}^{(j)}\right]$. The block vector and step updates are given by

$$\mathbf{x}_{i+1}^{(j)} := \mathbf{x}_i^{(j)} + \boldsymbol{E}_i d_i(\mathbf{x}_i^{(j)}, \alpha_i), \tag{27a}$$

$$\mathbf{x}^{(j+1)} := \mathbf{x}^{(j)} + \sum_{i=1}^{p} \boldsymbol{E}_i d_i(\mathbf{x}_i^{(j)}, \alpha_i). \tag{27b}$$

where $d_i(\mathbf{x}^{(j)}, \alpha_i)$ now satisfies the Armijo descent condition

$$\mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}_{i+1}^{(j)}) \geq -\sigma \nabla_i \mathcal{L}(\mathbf{x}_i^{(j)})^T d_i(\mathbf{x}_i^{(j)}, \alpha_i). \tag{28}$$

Similar to Section 4.1, the same block-Lipschitz continuous property holds Definition 4.1 for different Lipschitz constants, as does Proposition 4.1 for new α_i. Since every $d_i(\mathbf{x}_i^{(j)}, \alpha_i)$ is a descent direction for the ith block, $\cos(\theta) \geq 0$ as given by (24). Because we are not stationary, at least one of these directions will be nonzero.

Lemma 4.2 *[Cyclic Block PG Step Satisfies Sufficient Decrease] Let* $\{\mathbf{x}^{(j)}\}_{j\geq 0}$ *be the sequence generated by cycling through PG steps* (27). *Then for every* $j = 0, 1, 2, \ldots,$

$$\mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}^{(j+1)}) \geq \tilde{\sigma} \omega_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)}) \tag{29}$$

for $\tilde{\sigma} \in (0, 1)$

Proof Since we are cycling through blocks, we define each ith PG step satisfies (28) and also produces an angle

$$\cos\theta = \frac{-\sigma \nabla \mathcal{L}(\mathbf{x}_i^{(j)})^T \boldsymbol{E}_i d_i(\mathbf{x}_i^{(j)}, \alpha_i)}{\left\|\nabla \mathcal{L}(\mathbf{x}_i^{(j)})\right\| \left\|d_i(\mathbf{x}_i^{(j)}, \alpha_i)\right\|} \geq \frac{-\sigma \nabla_i \mathcal{L}(\mathbf{x}_i^{(j)})^T d_i(\mathbf{x}_i^{(j)}, \alpha_i)}{L \left\|d_i(\mathbf{x}_i^{(j)}, \alpha_i)\right\|} \geq \delta_i \geq 0.$$

Because we are not stationary, there exists at least one block where $\delta_i > 0$. For each i, we have that

$$\begin{aligned} \mathcal{L}(\mathbf{x}_i^{(j)}) - \mathcal{L}(\mathbf{x}_{i+1}^{(j)}) &\geq -\sigma \nabla \mathcal{L}(\mathbf{x}_i^{(j)})^T \boldsymbol{E}_i^T d_i(\mathbf{x}_i^{(j)}, \alpha_i) \\ &\geq \sigma L \delta_i \alpha_i \left(\alpha_i^{-1} \left\|d_i(\mathbf{x}_i^{(j)}, \alpha_i)\right\|\right). \end{aligned}$$

Summing over $i = 1, \ldots, p$, we have that

$$\mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}^{(j+1)}) \geq \sigma L \sum_{i=1}^{p} \frac{\delta_i \alpha_i}{\alpha_i} \left\|d_i(\mathbf{x}_i^{(j)}, \alpha_i)\right\|. \tag{30}$$

Utilizing a similar proof technique as in [5, Lemma 3.3], we have that for all $i = 1, \ldots, p$,

$$\begin{aligned}
\left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\| &\leq \left\| d_i(\mathbf{x}^{(j)}, \alpha_i) - d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| \\
&\leq \left\| \underset{\mathcal{X}_i}{\operatorname{proj}} \left(x_i^{(j)} - \alpha_i \nabla_i \mathcal{L}(\mathbf{x}^{(j)}) \right) - \underset{\mathcal{X}_i}{\operatorname{proj}} \left(x_i^{(j)} - \alpha_i \nabla_i \mathcal{L}(\mathbf{x}_i^{(j)}) \right) \right\| \\
&\quad \ldots + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| \\
&\leq \left\| x_i^{(j)} - x_i^{(j)} + \alpha_i (\nabla_i \mathcal{L}(\mathbf{x}_i^{(j)}) - \nabla_i \mathcal{L}(\mathbf{x}^{(j)})) \right\| + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| \\
&\leq \left\| \nabla_i \mathcal{L}(\mathbf{x}^{(j)}) - \nabla_i \mathcal{L}(\mathbf{x}_i^{(j)}) \right\| + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| \\
&\leq \left\| \nabla \mathcal{L}(\mathbf{x}^{(j)}) - \nabla \mathcal{L}(\mathbf{x}_i^{(j)}) \right\| + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\|,
\end{aligned}$$

where we use the Cauchy–Schwarz inequality and nonexpansivity of the projection operator, noting that $\alpha_i \leq 1$. We can use Lipschitz continuity to further reduce the inequality to the sum of the steps

$$\begin{aligned}
\left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\| &\leq L \left\| \mathbf{x}^{(j)} - \mathbf{x}_i^{(j)} \right\| + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| \\
&\leq L \left\| \sum_{l=1}^{i} \boldsymbol{E}_l d_l(\mathbf{x}_l^{(j)}, \alpha_\ell) \right\| + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| \\
&\leq L \left(\sum_{l=1}^{i} \left\| \boldsymbol{E}_l d_l(\mathbf{x}_l^{(j)}, \alpha_\ell) \right\| \right) + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\|,
\end{aligned}$$

again utilizing the Cauchy–Schwarz inequality and (27). Simplifying, we get

$$\left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\| \leq L \left(\sum_{l=1}^{i} \left\| d_l(\mathbf{x}_l^{(j)}, \alpha_\ell) \right\| \right) + \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\|. \tag{31}$$

Now we sum (31) over p with the necessary coefficients

$$\begin{aligned}
\sum_{i=1}^{p} \frac{\delta_i \alpha_i}{\alpha_i} \left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\| &\leq \sum_{i=1}^{p} \frac{\delta_i \alpha_i}{\alpha_i} \left(1 + (p + 1 - i) L \right) \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| \\
&\leq (1 + L(p+1)) \sum_{i=1}^{p} \frac{\delta_i \alpha_i}{\alpha_i} \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\|
\end{aligned}$$

where the latter inequality comes from (23). We can combine this with (30)

$$\begin{aligned}\mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}^{(j+1)}) &\geq \sigma L \sum_{i=1}^{p} \frac{\delta_i \alpha_i}{\alpha_i} \left\| d_i(\mathbf{x}_i^{(j)}, \alpha_i) \right\| \\ &\geq \frac{\sigma L}{1 + L(p+1)} \sum_{i=1}^{p} \frac{\delta_i \alpha_i}{\alpha_i} \left\| d_i(\mathbf{x}^{(j)}, \alpha_i) \right\| \\ &\geq \frac{\sigma L}{1 + L(p+1)} \sum_{i=1}^{p} \delta_i \alpha_i \left\| \operatorname*{proj}_{\mathcal{X}_i}(x_i - \nabla_i \mathcal{L}(\mathbf{x}^{(j)})) - x_i \right\| \\ &\geq \frac{\sigma L \delta}{1 + L(p+1)} \omega_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)})\end{aligned}$$

where, assuming we are not at a stationary point,

$$0 < \delta \leq \frac{\left(\sum_{i=1}^{p} \delta_i \alpha_i \left\| d_i(\mathbf{x}^{(j)}, 1) \right\|\right)}{\left(\sum_{i=1}^{p} \left\| d_i(\mathbf{x}^{(j)}, 1) \right\|\right)},$$

i.e. a value less than the fraction of PG steps not stationary (as δ_i may be zero for some blocks and $\delta_i, \alpha_i \leq 1$ by definition). If $\delta_i = 0$ for all i, then we are at a stationary point. Similar to Lemma 4.1,

$$\frac{\sigma L \delta}{1 + L(p+1)} \geq \frac{2\delta\sigma(1-\sigma)}{1 + L(p+1)} = \tilde{\sigma}$$

which again shows $\tilde{\sigma} \in (0, .5)$ for the maximum $\delta = 1$ and denominator greater than 1. Plugging this in yields (30).

4.3 Cyclic Minimization

If one wants to minimize in each coordinate, modifying Lemma 4.1 requires the method of choice to satisfy block sufficient decrease, akin to what the line search directly gives in (22). Examples of such may be with [49, 54], L-BFGS [46], or trust-region methods; these will also suffice in Line 10 of Algorithm 2. In fact, we assume that any algorithm used to produce a minimum satisfies some notion of sufficient decrease as in (13), but for each block:

$$\mathcal{L}(\mathbf{x}_i^{(j)}) - \mathcal{L}(\mathbf{x}_{i+1}^{(j)}) \geq \sigma \left\| \operatorname*{proj}_{\mathcal{X}_i}(x_i^{(j)} - \nabla \mathcal{L}(\mathbf{x}_i^{(j)})) - x_i^{(j)} \right\| \geq 0. \tag{32}$$

Now, we utilize the update rule

$$x_i^{(j+1)} \in \operatorname*{arg\,min}_{x \in \mathcal{X}_i} \mathcal{L}([\mathbf{x}_{i<}^{(j+1)}, x, \mathbf{x}_{>i}^{(j)}]), \tag{33a}$$

$$\mathbf{x}^{(j+1)} := \sum_{i=1}^{p} \boldsymbol{E}_i x_i^{(j+1)} = \mathbf{x}^{(j)} + \sum_{i=1}^{p} \boldsymbol{E}_i (x_i^{(j+1)} - x_i^{j}), \tag{33b}$$

where again $\mathbf{x}_i^{(j)}$ is represents the primal variables updated up to the ith block $\mathbf{x}_i^{(j)} = [\mathbf{x}_{<i}^{(j+1)}, x_i^{(j)}, \mathbf{x}_{i>}^{(j)}]$ with $\mathbf{x}_{i+1}^{(j)} = [\mathbf{x}_{<i}^{(j+1)}, x_i^{(j+1)}, \mathbf{x}_{i>}^{(j)}]$. From this, one can prove sufficient decrease in the same manner as Lemma 4.2.

Lemma 4.3 *[Cyclic Block Minimization] Let* $\{\mathbf{x}^{(j)}\}_{j\geq 0}$ *be the sequence generated by cycling through minimizing in each coordinate in* (33) *such that* (32) *holds. Then for every* $j = 0, 1, 2, \ldots,$

$$\mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}^{(j+1)}) \geq \tilde{\sigma}\omega_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)}) \tag{34}$$

Proof Cycling through the block yields

$$\begin{aligned} \mathcal{L}(\mathbf{x}^{(j)}) - \mathcal{L}(\mathbf{x}^{(j+1)}) &\geq \sigma \sum_{i=1}^{p} \left\| \underset{\mathcal{X}_i}{\operatorname{proj}}(x_i^{(j)} - \nabla_i \mathcal{L}(\mathbf{x}_i^{(j)})) - x_i^{(j)} \right\| \\ &\geq \sigma \sum_{i=1}^{p} \left\| \underset{\mathcal{X}_i}{\operatorname{proj}}(x_i^{(j)} - \alpha_i \nabla_i \mathcal{L}(\mathbf{x}_i^{(j)})) - x_i^{(j)} \right\|. \end{aligned} \tag{35}$$

where $\alpha_i \leq 1$ is an Armijo linesearch parameter and last inequality results from (23) in each block. Let $\bar{x}_i = \operatorname{proj}_{\mathcal{X}_i}(x_i^{(j)} - \alpha_i \nabla_i \mathcal{L}(\mathbf{x}_i^{(j)}))$. Using the same tricks as Lemma 4.2, we have

$$\begin{aligned} \left\| \underset{\mathcal{X}_i}{\operatorname{proj}}(x_i^{(j)} - \alpha_i \nabla_i \mathcal{L}(\mathbf{x}^{(j)})) - x_i^{(j)} \right\| &\leq L \left\| \mathbf{x}^{(j)} - \mathbf{x}_i^{(j)} \right\| + \left\| \bar{x}_i - x_i^{(j)} \right\| \\ &\leq L \sum_{\ell=1}^{i-1} \left\| x_\ell^{(j)} - x_\ell^{(j+1)} \right\| + \left\| \bar{x}_i - x_i^{(j)} \right\|. \end{aligned}$$

If $\left\| x_\ell^{(j)} - x_\ell^{(j+1)} \right\| \leq \left\| \bar{x}_\ell - x_\ell^{(j)} \right\|$ (i.e. the distance to the minima is less than the sufficient decrease condition for each block), then we can simplify the RHS by replacing $x_\ell^{(j+1)}$ with $\bar{x}_\ell$. Otherwise, note that because $\bar{x}_\ell - x_\ell^{(j)}$ and $x_\ell^{(j+1)} - x_\ell^{(j)}$ are descent directions by assumption (32), we can form a triangle between the three points; set the lengths of vectors $a = \left\| x_\ell^{(j)} - \bar{x}_\ell \right\|$, $b = \left\| x_\ell^{(j+1)} - \bar{x}_\ell \right\|$, and $c = \left\| x_\ell^{(j+1)} - x_\ell^{(j)} \right\|$, with $\theta_a, \theta_b, \theta_c$ the angles between b-c, a-c, and a-b respectively (i.e. θ_a faces a, etc.). We have that from the sine rule, $\frac{c}{\sin\theta_c} = \frac{a}{\sin\theta_a}$, or

$$\frac{\sin\theta_c}{\sin\theta_a} \left\| x_\ell^{(j)} - \bar{x}_\ell \right\| = \kappa_\ell \left\| x_\ell^{(j)} - \bar{x}_\ell \right\| = \left\| x_\ell^{(j+1)} - x_\ell^{(j)} \right\|,$$

which is guaranteed to be greater than zero unless we are stationary. Hence,

$$\left\| \underset{\mathcal{X}_i}{\operatorname{proj}}(x_i^{(j)} - \alpha_i \nabla_i \mathcal{L}(\mathbf{x}^{(j)})) - x_i^{(j)} \right\| \leq L \sum_{\ell=1}^{i-1} \left(\kappa_\ell \left\| \bar{x}_\ell - x_\ell^{(j)} \right\| \right) + \left\| \bar{x}_i - x_i^{(j)} \right\|$$

Proceeding in a similar manner as Lemma 4.2, we have

$$\begin{aligned}\sum_{i=1}^{p}\left\|\operatorname*{proj}_{\mathcal{X}_i}(x_i^{(j)}-\alpha_i\nabla_i\mathcal{L}(\mathbf{x}^{(j)}))-x_i^{(j)}\right\| &\leq \sum_{i=1}^{p}\left\|\bar{x}_i-x_i^{(j)}\right\|+L\sum_{\ell=1}^{i-1}\kappa_\ell\left\|\bar{x}_\ell-x_\ell^{(j)}\right\| \\ &= \sum_{i=1}^{p}(1+(p-i)L\kappa_i)\left\|\bar{x}_i-x_i^{(j)}\right\| \\ &\leq (1+Lp\kappa_{\max})\sum_{i=1}^{p}\left\|\bar{x}_i-x_i^{(j)}\right\|,\end{aligned}$$

where $\kappa_{\max}=\max_{i=1,\dots,p}\kappa_i$ the maximum angle ratio. We acquire (34) by plugging the above into (35)

$$\begin{aligned}\mathcal{L}(\mathbf{x}^{(j)})-\mathcal{L}(\mathbf{x}^{(j+1)}) &\geq \frac{\sigma}{1+Lp\kappa_{\max}}\sum_{i=1}^{p}\frac{\alpha_i}{\alpha_i}\left\|\operatorname*{proj}_{\mathcal{X}_i}(x_i^{(j)}-\alpha_i\nabla_i\mathcal{L}(\mathbf{x}^{(j)}))-x_i^{(j)}\right\| \\ &\geq \frac{\sigma}{1+Lp\kappa_{\max}}\sum_{i=1}^{p}\alpha_i\left\|\operatorname*{proj}_{\mathcal{X}_i}(x_i^{(j)}-\nabla_i\mathcal{L}(\mathbf{x}^{(j)}))-x_i^{(j)}\right\| \\ &\geq \frac{\sigma\delta}{1+Lp\kappa_{\max}}\left\|\sum_{i=1}^{p}\operatorname*{proj}_{\mathcal{X}_i}(x_i^{(j)}-\nabla_i\mathcal{L}(\mathbf{x}^{(j)}))-x_i^{(j)}\right\| \\ &\geq \tilde{\sigma}\omega_{\rho^{(k)}}(\mathbf{x}^{(j)},\mathbf{y}^{(k)})\end{aligned}$$

where, similar to lemma 4.2,

$$0<\delta\leq\frac{\left(\sum_{i=1}^{p}\alpha_i\left\|d_i(\mathbf{x}^{(j)},1)\right\|\right)}{\left(\sum_{i=1}^{p}\left\|d_i(\mathbf{x}^{(j)},1)\right\|\right)}\leq 1,$$

and $\tilde{\sigma}=\frac{\sigma\delta}{1+Lp\kappa_{\max}}<1$ for $\sigma\in(0,1)$.

Remark 4.1 We offer several schemes, all of which achieve the decrease required by the filter; one can be flexible in utilizing these throughout any part of the filter loop. While Lemma 4.3 is perhaps the closest to "true" ADMM, we see numerically that such a scheme is slow, especially early in the algorithm; this may stem from the nature of running ADMM on a nonconvex problem. In the numerical experiments below, we find that using Lemma 4.2 for (13) in 6 of Algorithm 2 results in the fastest convergence. We leave explorations into acceleration for future work.

5 Nonconvex Bilinear Optimization: Nonnegative Matrix Factorization and Completion

The well-known nonnegative matrix factorization and completion (NMF/C) problem extracts two factors $X\in\mathbb{R}^{N\times K}, Y\in\mathbb{R}^{K\times Q}$ from a potentially restricted observation or data matrix $M\in\mathbb{R}^{N\times Q}$. The most popular form [33] of this problem is given by

$$\min_{X,Y} \frac{1}{2} \left\| \mathcal{A}_{\mathcal{S}}(XY - M) \right\|_F^2 \quad \text{s.t. } X, Y \geq \mathbf{0}, \tag{36}$$

where $\mathcal{S} \subset \{1, 2, \ldots, n\}$ contains the indices of known entries and $\mathcal{A}_S$ (recall (9)) denotes the projection onto the observed set S. Here, the two factors X, Y have rank $K \ll \min(Q, N)$, and both $X, Y \geq \mathbf{0}$. NMF/C has many practical applications, such as text mining, pattern discovery, bioinformatics, and clustering (see, e.g., [6, 30, 33, 34]). Plenty of algorithms have been proposed for NMF/C; ADMM applied to NMF has been described by [10], with more specialized algorithms developed in [1, 27, 32, 52]. While, (36) is nonconvex, introducing $Z \in \mathbb{R}^{N\times Q}$ and $W \in \mathbb{R}^{N\times Q}$ and setting $Z = XY$ as a constraint transforms each block update of the augmented Lagrangian into a simple convex problem, at the expense of increased dimensionality and bilinear constraints:

$$\min_{X,Y,Z,W} \frac{1}{2} \left\| Z - W \right\|_F^2 \quad \text{s.t. } X, Y \geq \mathbf{0},\ Z = XY,\ \mathcal{A}_S(W - M) = \mathbf{0}, \tag{37}$$

where $W = M$ if $\mathcal{S} \equiv \{1, \ldots, n\}$. An ADMM approach for (36) was given by [10], and a bilinear approach was given in [30]. The authors of [30] show that their scheme produces iterates that converge for $\rho > 1$, ensuring satisfaction of the primal gap in the limit. Our algorithm convergence framework encompasses this result; if the initialization of the penalty parameter does not necessitate restoration, our algorithm may default to standard ADMM, depending on filter acceptance. Our numerical experiments show that Algorithm 2 can also update poor ρ initializations.

5.1 ADMM-Filter Restoration-step Details for NMF

Problem (37) elicits the η and ω definitions

$$\eta(\mathbf{x}) := \left\| Z - XY \right\|_F, \text{ and } \omega_\rho(\mathbf{x}, \mathbf{y}) := \left\| \operatorname*{proj}_{\mathcal{X}} \left(\mathbf{x} - \nabla \mathcal{L}_{\rho^{(k)}}(\mathbf{x}, \mathbf{y}) \right) - \mathbf{x} \right\|_F. \tag{38}$$

Since the process of feasibility restoration in Algorithm 2 is flexible, we define a linesearch routine adapted for NMF/C that exploits the structure of (37) and prove that it always produces a feasible point. Restoration is triggered by conditions (14a) and (14b), where either $\eta^{(j)} = \left\| Z^{(j+1)} - X^{(j+1)} Y^{(j+1)} \right\| \geq U_{\text{NMF}} := \max(\omega_{\min}/\gamma, \beta\eta_{\min})$, or $\omega_{\min} \leq \epsilon$ and $\eta^{(j)} > \beta\eta_{\min}$. As in [50], $\omega_{\min} := \min\{\omega^{(l)} : (\eta^{(l)}, \omega^{(l)}) \in \mathcal{F}\}$, with $\eta_{\min}$ being the corresponding η value. Assume that $Z^{(j+1)} \neq X^{(j+1)} Y^{(j+1)}$ is infeasible, not filter acceptable, and this iteration invokes the feasibility restoration phase. The potential Z update $\tilde{Z} = X^{(j+1)} Y^{(j+1)}$ is feasible and filter acceptable as $\eta^{(j)}$ would equal zero (note the filter does not accept $\eta = 0$ points). Our "new" restoration filter points is a function of α along direction $\mathbf{d} = [\mathbf{0}; \mathbf{0}; (\tilde{Z} - Z^{(j+1)})]$

$$\mathbf{x}^{(j+1)}(\alpha) = \mathbf{x}^{(j+1)} + \alpha \mathbf{d} = \left[X^{(j+1)}; Y^{(j+1)}; Z^{(j+1)} + \alpha(\tilde{Z} - Z^{(j+1)}) \right].$$

such that our feasibility is now

$$\eta(\mathbf{x}^{(j+1)}(\alpha)) := \left\| Z^{(j+1)} - \alpha(\tilde{Z} - Z^{(j+1)}) - X^{(j+1)}Y^{(j+1)} \right\|_F , \tag{39}$$

and likewise for first-order error $\omega_{\rho^{(k)}}(\mathbf{x}^{(j+1)}(\alpha)$.

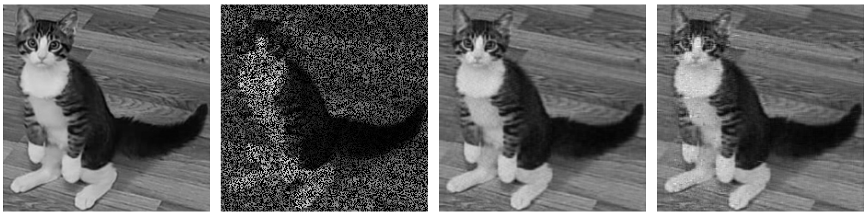
Lemma 5.1 (Existence of a Filter Acceptable Point for NMF with Bilinear Constraints) *At a filter-unacceptable point given in Algorithm 2 and $\eta(\mathbf{x}^{(j+1)}(\alpha))$ given by* (39)*, there exists an $\alpha \in (0, 1)$ such that $\mathbf{x}^{(j+1)}(\alpha)$ is a filter-acceptable point.*

Proof By definition of η in (39) and $\mathbf{c}(\mathbf{x}^{(j+1)}) = Z^{(j+1)} - X^{(j+1)}Y^{(j+1)}$, we have that the feasibility is a continuous function and $\eta = 0$ if and only if $Z^{(j+1)} = X^{(j+1)}Y^{(j+1)}$. The feasibility metric $\eta(\mathbf{x}^{(j+1)}(\alpha))$ is monotonically decreasing as $\alpha \in [0, 1]$ increases from 0 to 1, as

$$\frac{\partial}{\partial \alpha} \left\| Z + \alpha(XY - Z) - XY \right\|_F = (\alpha - 1) \frac{\left\| XY - Z \right\|_F^2}{\left\| (\alpha - 1)(XY - Z) \right\|_F} \leq 0$$

with $\eta(\mathbf{x}^{(j+1)}(1)) = 0$. Note $\alpha = 1$ ($\eta = 0$) points are not added to the filter; we instead wish to select the smallest α that provides filter acceptability, because other choices may be far from the current iterate. When $\alpha = 0$, we are not feasible or filter acceptable by (14a) and (14b). By the intermediate value theorem and monotonicity of η, there exists an $\alpha^* \in (0, 1)$ such that $\eta(\mathbf{x}^{(j+1)}(\alpha^*)) = \beta\eta^* > 0$. Since $\eta(\mathbf{x}^{(j+1)}(\alpha^* + \varepsilon)) < \beta\eta^*,\ \forall \varepsilon \in (0, 1 - \alpha^*]$, our linesearch is guaranteed to create a filter acceptable point if we choose any $\alpha > \alpha^*$. □

Such an α would yield $\eta^{(j)} < \beta\eta^*$, where η^* is the smallest η such that $(\eta, \omega) \in \mathcal{F}$. We run a bisection method on $\eta(\mathbf{x}^{(j+1)}(\alpha))$ until we reach filter acceptance guaranteed by Lemma 5.1.



(a) M - 100% (b) M - 50% (c) XY - 100% (d) XY - 50%

Fig. 1 NMF/C solution for Sylvester with full and 50% coverage; XY distinctions are minimal.

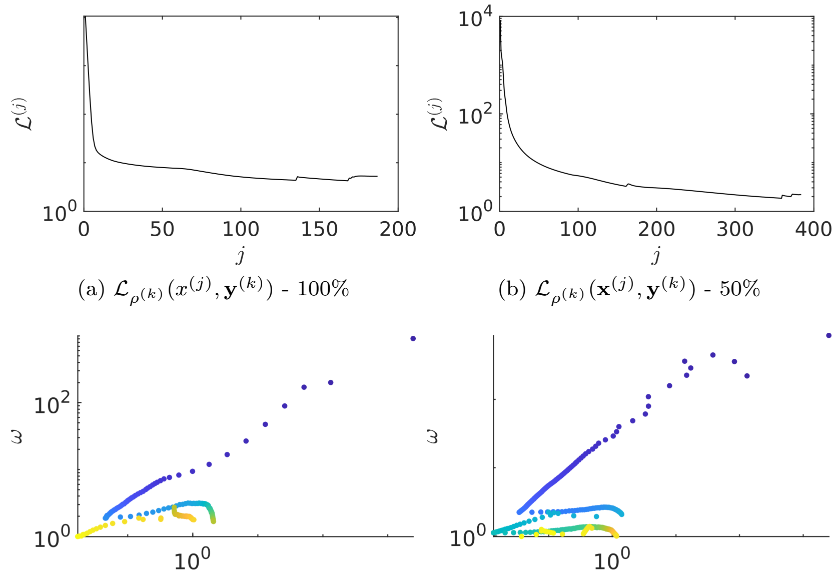
5.2 NMF/C Numerical Results

While Problem (37) allows for simultaneous $(X, Z), (Y, W)$ updates, we separate them for our experiments as we saw no significant performance benefits either way. Our data matrix is the image $M \in \mathbb{R}^{225\times225}$ in Figure 1a with added noise from $\mathcal{N}(0, .01)$, and the solution is $X \in \mathbb{R}^{225\times45}, Y \in \mathbb{R}^{45\times225}$. Our goal is to recapitulate it with 5 times the compression in matrix rank, with 100% coverage and 50% coverage (i.e. 50% missing pixels). We start our algorithm at a random point satisfying the bounds $X^{(0)}, Y^{(0)} \geq \mathbf{0}$, $W^{(0)} = \mathcal{A}_{\mathcal{S}}(M)$, and $Z^{(0)} = \mathbf{0}$ with $\beta = .9$, $\gamma = 1 - \beta$, $\epsilon = 10^{-1}$. The first-order restoration switching condition (14b) is triggered when $\omega_0 \leq 10^{-3}$. The initial penalty parameter is set to be $\rho^{(0)} = 1.1$ from [30]. We allow for a maximum of 200 outer iterations and 200 inner iterations, but note that neither are attained in this experiment. The block minimization utilizes L-BFGS via `MinConf_SPG` [46], with a maximum of 100 iterations and optimality criteria of 10^{-5}. The algorithm exits when the relative first-order error and feasibility are less than 10^{-3} and the absolute first-order error and feasibility are less than 1. The compressed reconstruction of the chosen image is given in Figure 1. Figure 2 displays augmented Lagrangian decrease, and the total filter entries with blue being earlier entries and yellow being late iteration entries; observe that Figure 2c depicts the filter guiding the iterates toward the origin. We also note that the ρ value did not change.

Next, we conduct a study to examine how Algorithm 2 picks up on poor initial penalty parameters. We initialized the algorithm seven times with the same initial conditions as above but seven different penalty parameters: $\rho^{(0)} \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\}$. From [30], we expect $\rho^{(0)} > 1$ to be sufficient. We see from Figure 3a that Algorithm 2 has been able pick up that $\rho^{(0)} = \{10^{-3}, 10^{-2}\}$ were very poor initial guesses for both coverages, triggering early restoration. From Figure 3b we can see that $\rho^{(0)} = 10^{-1}$ performed well in minimizing the augmented Lagrangian; it was not theoretically guaranteed to converge in the ADMM algorithm presented in [30]. These results indicate that Algorithm 2 can successfully identify insufficient penalty parameters and can converge with poor ρ choices. One downside is the length of time needed to determine insufficient penalty parameter values; addressing this downside remains future work.

6 Chemical Spectrum Analysis

The NMF/C example can be modified to solve a more complex physical problem, namely finding distributions of chemicals that occur in measured spectrum analysis. We employ a simplified analysis and assume that each chemical concentration follows a Gaussian distribution, and ascertain the nonnegative combination of Gaussians that reproduce the spectra data by framing the problem through the nonlinear NMFC lens.



(a) $\mathcal{L}_{\rho^{(k)}}(x^{(j)}, \mathbf{y}^{(k)})$ - 100%

(b) $\mathcal{L}_{\rho^{(k)}}(\mathbf{x}^{(j)}, \mathbf{y}^{(k)})$ - 50%

(c) Total filter entries - 100%

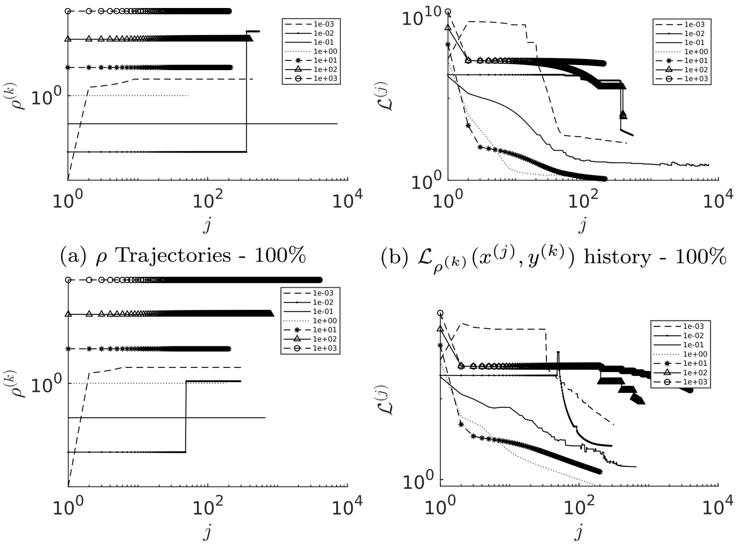
(d) Total filter entries - 50%

Fig. 2 Descent and filter stats for Sylvester with full and 50% coverage; there are two and one points in the final filter, respectively. Blue points are earlier, yellow points are later entries.

We let μ_k and σ_k be the mean and standard deviation of the kth Gaussian, respectively, with μ and σ the vectors of all K Gaussian moments. The *intensity function* $\hat{I}(w, c; \mu, \sigma)$ is a function of wave number w, concentration c, and the Gaussian moments. Given $m = 22$ number of concentrations and scaled wavenumbers $w_i = 1, \dots 1750$, we have a data matrix $M = \left[\hat{I}(w, c, \mu, \sigma)\right] \in \mathbb{R}^{1750 \times 22}$. We note here that this K does not necessarily perform the function of a compression variable; rather, it is indicative of the number of chemical peaks. Each chemical peak is centered around a (scaled) wavenumber, which may span from $n = 0, \dots, 1750$; these make up the spectra. The number of chemicals are within a particular sample may be unknown, and therefore K may be greater than the rank of the data matrix.

Instead of the nominal static matrix variables X, Y, Z and data matrix M, these are now nonlinear functions dependent on wave numbers w and concentrations c, and moments μ and σ. Our intensity function is reformulated as a summation of coefficients based on concentration $Y_{k,\ell} = f_k(c_\ell)$ affecting our Gaussians dependent on wavenumber $g_k(w_i, \mu_k, \sigma_k) = \exp\left(-\frac{1}{2}\left(\frac{w_i - \mu_k}{\sigma_k}\right)^2\right)$:

$$\hat{I}(w_i, c_\ell, \mu, \sigma) = \sum_{k=1}^{K} \exp\left(-\frac{1}{2}\left(\frac{w_i - \mu_k}{\sigma_k}\right)^2\right) Y_{k,\ell} = X_{i,:} Y_{:,\ell} \qquad (40)$$



(a) ρ Trajectories - 100%

(b) $\mathcal{L}_{\rho^{(k)}}(x^{(j)}, y^{(k)})$ history - 100%

(c) ρ Trajectories - 50%

(d) $\mathcal{L}_{\rho^{(k)}}(x^{(j)}, \mathbf{y}^{(k)})$ history - 50%

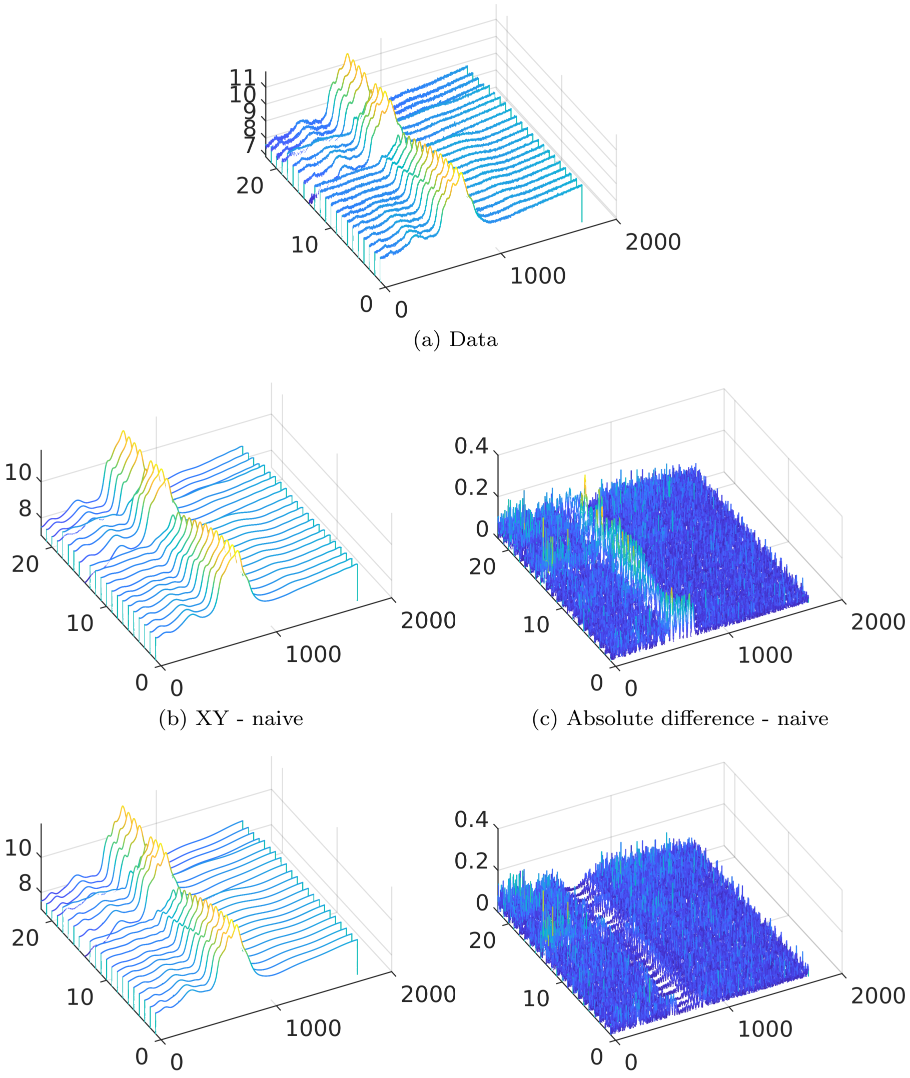
Fig. 3 Convergence metrics for different ρ initializations for full and 50% coverage.

We seek to represent $\left[\hat{I}(w, c, \mu, \sigma)\right] = X(\mu, \sigma)Y$ as in (40) and the observed intensity data by $M \in \mathbb{R}^{m\times n}$. We withhold 10% of the data for testing, constructing the NMFC problem

$$\min_{\mu,\sigma,Y,Z} \|\mathcal{A}_{\mathcal{S}}(M - Z)\|^2 \quad \text{s.t.} \;\; \sigma,\, Y \geq 0,\; Z = X(\mu, \sigma)Y, \tag{41}$$

where again $X(\mu, \sigma) \in \mathbb{R}^{m\times K}$ are the Gaussians, $Y \in \mathbb{R}^{K\times n}$ are the weights, and $\mathcal{S}$ is a random index containing 90% of total concentrations. We start with $K = 22, m = 22, n = 1750$. The intensity M data is given by Figure 4a. We use `MinConf_SPG`[46] with optimality 10^{-5} and maximum iterations are 1000. We utilize the same algorithmic parameters as in the NMF/C examples with maximum outer iteration of 1000 and maximum filter iteration of 100. Our absolute stopping tolerances of $\epsilon = 10^{-1}$ and our initial penalty parameter is $\rho^{(0)} = 4.0$ for all experiments. Our naive K initialization spaces μ evenly throughout the n spectra and all $\sigma = n/K$. We initially set $Y = \mathbf{0}$ and warm-start (41) with $Y \leftarrow \arg\min_{Y\geq 0} \|X(\mu, \sigma)Y - M\|_F^2$.

The solution for this initialization with $K = 22$ is given in Figure 4b with absolute difference in Figure 4c, in which we observe that the reconstruction captures the data fairly accurately, with more error where the intensity peaks are located. We can improve upon the naive initialization performance by instead conducting a parameter sweep to determine the most effective K. Starting from $K = 5$, we minimize and incrementally increase K by putting a new Gaussian mean at the location of highest error obtained from the previous value of K. We run the experiments for $K = \{5, \ldots, 33\}$, and again plot $K = 22$ in Figure 4. Figures 4d and 4e depict the results, which show



(a) Data

(b) XY - naive

(c) Absolute difference - naive

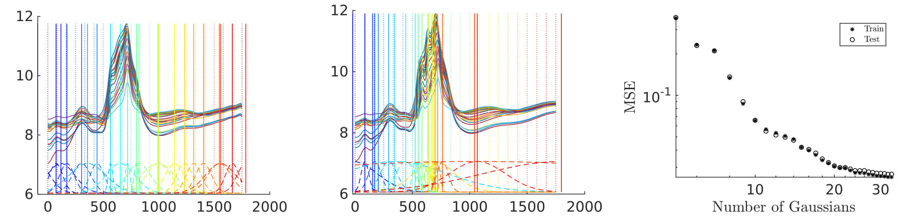
(d) XY - sequential

(e) Absolute difference - sequential

Fig. 4 Spectra reconstruction via Algorithm 2 with 22 Gaussians. All graphs are wavelength (x-axis) by concentration (y-axis) by intensity (z-axis).

improvement in capturing wavelength where the signal is the strongest. To further illustrate how the Gaussian distributions create the intensity and for comparison between naive and sequential initializations, Figures 5a and 5b take a 2D view of Figures 4b and 4d along the y-axis and also plot the recovered Gaussian moments along the wavelength to see where they align with the intensity. From this, we can observe how the different initializations allow for different moment drifts across the spectra. The mean-squared error in

Figure 5c shows that fit improves as K increases, but such improvement is marginal past $K > 22$.



(a) Naive (b) Sequential (c) MSE by K.

Fig. 5 Figures 5a and 5b show 2D spectra plotted with distributions along the wavelength axis along with their respective mean given by the solid line; intensity, $\hat{I}$ is on the y-axis. We additionally plot naive initialization as dotted lines to show Gaussian moment drift. Figure 5c the mean-squared error value per number of Gaussians

7 Conclusion

We present a convergent fully block-separable ADMM-filter algorithm that solves difficult constrained nonconvex problems. We demonstrated algorithm effectiveness on highly nonlinear objectives in the NMF/C realm. In addition to showing convergence for ADMM, we depicted the filter's ability to correct poor initial penalty parameter choices. Next steps entail generalizing Section 4 to nonsmooth regularizers and subsequently generalizing the filter convergence proof. Additionally, current methodology allows only for ρ increases, but some work has been done in penalty parameter "acceleration" [53, 55]; we leave this examination for future work.

Statements and Declarations

Competing Interests & Funding: The authors declare that they have no conflicts of interest.

This research was sponsored by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research's John von Neumann Fellowship. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. Research partially supported by DOE CSGF grant number DE-FG02-97ER25308. This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research's applied mathematics program and Office of Basic Energy Sciences' Separation Science program under Contract Nos. DE-AC02-05CH11231 and DE-AC02-06CH11357. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Data Availability: All datasets generated for the numerical results are available from the corresponding author upon reasonable request.

References

1. A. M. S. Ang and N. Gillis. Accelerating nonnegative matrix factorization algorithms using extrapolation. *Neural Computation*, 31(2):417–439, 2019. doi: 10.1162/neco_a_01157.
2. H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka–Lojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.
3. H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, volume 408. Springer, 2017.
4. A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974997.
5. A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal of Optimization*, 23(4), 2013. doi: 10.1137/120887679.
6. M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, 2007. doi: 10.1016/j.csda.2006.11.006.
7. D. Bertsekas. *Nonlinear Programming*. Athena scientific optimization and computation series. Athena Scientific, 2016. ISBN 9781886529052.
8. D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989. ISBN 0-13-648700-9.
9. R. I. Boţ and D.-K. Nguyen. The proximal alternating direction method of multipliers in the nonconvex setting: Convergence analysis and rates. *Mathematics of Operations Research*, 45(2):682–712, 2020.
10. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
11. C. Chen, Y. Ye, B.-S. He, and X. Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155:57–79, 2016.
12. L. Chen, D. Sun, and K.-C. Toh. An efficient inexact symmetric Gauss–Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Mathematical Programming*, 161:237–270, 2017.
13. C. M. Chin and R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Mathematical Programming*, 96(1):161–177, 2003.
14. A. R. Conn, N. I. M. Gould, and P. L. Toint. *LANCELOT: a Fortran package for large-scale nonlinear optimization (release A)*. Springer, Heidelberg, 2003.
15. D. Davis and W. Yin. Convergence rate analysis of several splitting schemes. In W. Yin, S. Osher, and R. Glowinski, editors, *Splitting Methods in Communication, Imaging, Science, and Engineering*, pages 115–163. Scientific Computation. Springer, Cham, 2016.
16. W. Deng, M.-J. Lai, Z. Peng, and W. Yin. Parallel multi-block ADMM with o(1/k) convergence. *Journal of Scientific Computing*, 71:712–736, 2017.
17. E. D. Dolan, R. M. Lewis, and V. J. Torczon. On the local convergence of pattern search. *SIAM Journal on Optimization*, 14(2):567–583, 2003.
18. J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82:421–439, 1956.
19. J. Eckstein and D. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.
20. J. Eckstein and W. Yao. Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. *Pacific Journal on Optimization*, 11:619–644, 2015.
21. R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–270, 2002.
22. R. Fletcher, S. Leyffer, and P. L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM Journal of Optimization*, 13(1):44–59, 2002.

23. R. Fletcher, S. Leyffer, and P. L. Toint. A brief history of filter methods. Technical report, Argonne National Laboratories, 2006.
24. M. Friedlander. *A Globally Convergent Linearly Constrained Lagrangian Method for Nonlinear Optimization.* PhD thesis, Stanford University, 2002.
25. M. P. Friedlander and S. Leyffer. Global and finite termination of a two-phase augmented Lagrangian filter method for general quadratic programs. *SIAM Journal on Scientific Computing*, 30(4):1706–1729, 2008. doi: 10.1137/060669930.
26. D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2:17–40, 1975.
27. W. Gao, D. Goldfarb, and F. E. Curtis. ADMM for multiaffine constrained optimization. *Optimization Methods and Software*, 35(2):257–303, 2020. doi: 10.1080/10556788.2019.1683553.
28. R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problémes de Dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis*, 9(R2):41–76, 1975.
29. M. Gürbüzbalaban, A. Ozdaglar, N. D. Vanli, and S. J. Wright. Randomness and permutations in coordinate descent methods. *Mathematical Programming*, 181(2):349–376, 2020. doi: 10.1007/s10107-019-01438-4.
30. D. Hajinezhad, T.-H. Chang, X. Wang, Q. Shi, and M. Hong. Nonnegative matrix factorization using ADMM: Algorithm and convergence analysis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4742–4746. IEEE, 2016. doi: 10.1109/icassp.2016.7472577.
31. D. Han and X. Yuan. A note on the alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, 155:227–238, 2012.
32. L. T. K. Hien, D. N. Phan, and N. Gillis. A framework of inertial alternating direction method of multipliers for non-convex non-smooth optimization. *Computational Optimization and Applications*, Accepted, 2022.
33. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. doi: 10.1038/44565.
34. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.
35. H. Li and Z. Lin. Accelerated alternating direction method of multipliers: An optimal o(1/k) nonergodic analysis. *Journal of Scientific Computing*, 79:671–699, 2019.
36. M. Li, D. Sun, and K.-C. Toh. A convergent 3-block semi-proximal ADMM for convex minimization problems with one strongly convex block. *Asia-Pacific Journal of Operational Research*, 32(3):1550024, 2015.
37. T. Lin, S. Ma, and S. Zhang. On the global linear convergence of the ADMM with multiblock variables. *SIAM Journal on Optimization*, 25(3):1478–1497, 2015. doi: 10.1137/140971178.
38. T. Lin, S. Ma, and S. Zhang. On the sublinear convergence rate of multi-block ADMM. *Journal of the Operations Research Society of China*, 3:251–274, 2015. doi: 10.1007/s40305-015-0092-0.
39. T. Lin, S. Ma, and S. Zhang. Global convergence of unmodified 3-block ADMM for a class of convex minimization problems. *Journal of Scientific Computing*, 76:69–88, 2018. doi: 10.1007/s10915-017-0612-7.
40. P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16:964–979, 1979.
41. J. Liu and S. J. Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015. doi: 10.1137/140961134.
42. J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, second edition, 2006. doi: 10.1007/978-0-387-40065-5.
43. M. J. D. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4:193–201, 1973.
44. P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144:1–38,

2014. doi: 10.1007/s10107-012-0614-z.

45. R. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, 2009. doi: 10.1007/978-3-642-02431-3.
46. M. Schmidt, E. Berg, M. Friedlander, and K. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 456–463, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 April 2009. PMLR.
47. D. Sun, K.-C. Toh, and L. Yang. A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints. *SIAM Journal on Optimization*, 25(2):882–915, 2015. doi: 10.1137/140964357.
48. M. Sun and H. Sun. Improved proximal ADMM with partially parallel splitting for multi-block separable convex programming. *Journal of Applied Mathematics and Computing*, 58:151–181, 2018.
49. P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001. doi: 10.1023/a:1017501703105.
50. C. Vanaret and S. Leyffer. An augmented Lagrangian filter method. *Mathematical Methods of Operations Research*, 92(2):343–376, 2020. doi: 10.1007/s00186-020-00713-x.
51. J. J. Wang and W. Song. An algorithm twisted from generalized ADMM for multi-block separable convex minimization models. *Journal of Computational and Applied Mathematics*, 309:342–358, 2017. ISSN 0377-0427. doi: 10.1016/j.cam.2016.02.001.
52. Y. Wang, W. Yin, and J. Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78:29–63, 2018.
53. B. Wohlberg. ADMM penalty parameter selection by residual balancing. *arXiv preprint arXiv:1704.06209v1*, 2017.
54. Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal of Imaging Sciences*, 6(3):1758–1789, 2013.
55. Y. Xu, M. Liu, Q. Lin, and T. Yang. ADMM without a fixed penalty parameter: Faster convergence with new adaptive penalization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
56. L. Yang, D. Sun, and K.-C. Toh. A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions. *Mathematical Programming*, 155: 333–373, 2016.
57. X. Yuan, S. Zeng, and J. Zhang. Discerning the linear convergence of ADMM for structured convex optimization through the lens of variational analysis. *Journal of Machine Learning Research*, 21(83):1–75, 2020.
58. J. Zeng, S.-B. Lin, Y. Yao, and D.-X. Zhou. On ADMM in deep learning: Convergence and saturation-avoidance. *Journal of Machine Learning Research*, 22(199):1–67, 2021.