SIAM J. OPTIM.
Vol. 32, No. 3, pp. 2294–2318

A DIMENSION REDUCTION TECHNIQUE FOR LARGE-SCALE STRUCTURED SPARSE OPTIMIZATION PROBLEMS WITH APPLICATION TO CONVEX CLUSTERING*

YANCHENG YUAN†, TSUNG-HUI CHANG‡, DEFENG SUN†, AND KIM-CHUAN TOH§

Abstract. In this paper, we propose a novel adaptive sieving (AS) technique and an enhanced AS (EAS) technique, which are solver independent and can accelerate optimization algorithms for solving large-scale convex optimization problems with intrinsic structured sparsity. We establish the finite convergence property of the AS and EAS techniques with inexact solutions of the reduced subproblems. As an important application, we apply the AS and EAS techniques to the convex clustering model, which can accelerate the state-of-the-art algorithm SSNAL by more than 7 times and the algorithm ADMM by more than 14 times.

Key words. adaptive sieving, structured sparsity, dimension reduction, convex optimization, convex clustering

MSC codes. 90C06, 90C25, 90C90

DOI. 10.1137/21M1441080

1. Introduction.

Clustering is one of the core problems in data science. It plays an important role in numerous applications. Significant advances have been achieved in clustering during the last few decades, including K-means [18, 1], spectral clustering [23, 28], subspace clustering [29, 35], and so on. Despite these developments, some known drawbacks of these centroid based models are still challenging to overcome, such as sensitivity to the initialization, limited effectiveness in high dimensional problems, and the requirement on prior knowledge of the number of clusters. Here, we want to emphasize that the requirement on prior knowledge of the number of clusters is impractical for most real applications. Indeed, estimating the number of clusters itself is as challenging as clustering. One may argue that we can run classical clustering algorithms, such as K-means, with a few guesses on the number of clusters, but these clustering results are usually independent. Thus, users still need to determine the final clustering results subjectively based on their own preference.

The convex clustering model was proposed in [10, 17, 25]. Since then, it has become increasingly popular due to its good empirical performance and convincing theoretical guarantees [5, 13, 24, 30, 31, 42]. Specifically, for a given data matrix

*Received by the editors August 17, 2021; accepted for publication (in revised form) June 20, 2022; published electronically September 13, 2022.
https://doi.org/10.1137/21M1441080
Funding: The research of the first author is supported by the Hong Kong Polytechnic University under grant P0038284. The research of the second author is in part supported by the Shenzhen Research Institute of Big Data under grant 2019ORF01002. The research of the third author is supported by the Hong Kong Research Grants Council under grant 15303720 and by the Shenzhen Research Institute of Big Data under grant 2019ORF01002. The research of the fourth author is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 3 grant (MOE-2019-T3-1-010).

†Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (yancheng.yuan@polyu.edu.hk, defeng.sun@polyu.edu.hk).
‡School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), and Shenzhen Research Institute of Big Data, China (changtsunghui@cuhk.edu.cn).
§Department of Mathematics and Institute of Operations Research and Analytics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore (mattohkc@nus.edu.sg).

$A \in \mathbb{R}^{d\times N}$ of N data points, the convex clustering model is to solve the following optimization problem:

$$\min_{X\in\mathbb{R}^{d\times N}} \quad \frac{1}{2}\sum_{i=1}^{N}\|X_{:i}-A_{:i}\|^2+\lambda\sum_{1\le i<j\le N} w_{ij}\|X_{:i}-X_{:j}\|_q, \tag{1.1}$$

where $X_{:i}$ (or $A_{:i}$) is the ith column of X (or A), $w_{ij}=w_{ji}\ge 0$ are given weights, and $\lambda\ge 0$ is the hyperparameter to control the effect of the diffusion penalty. Here $\|\cdot\|_q$ is the vector q-norm, and we require $q\ge 1$ to guarantee the convexity of the model. After solving the model (1.1) with the optimal solution X^*, we assign the data points $A_{:i}$ and $A_{:j}$ to the same cluster if $X^*_{:i}=X^*_{:j}$. Readers who are interested in more details about cluster identification based on the convex clustering model with an inexact solution are referred to [4, 12, 30]. It has been proved in [4] that the convex clustering model (1.1) can generate a continuous clustering path with respect to the hyperparameter λ. Thus, prior knowledge on the number of clusters is *not* required. Instead, we will generate a clustering path by solving the convex clustering model (1.1) for a sequence of values of λ.

Motivated by the convex clustering model (1.1), in this paper, we consider structured optimization problems of the following form:

$$\min_{x\in\mathbb{R}^n} \quad F_\lambda(x):=f(x)+\lambda p(Bx), \tag{1.2}$$

where $\lambda>0$ is a hyperparameter, $f:\mathbb{R}^n\to\mathbb{R}$ is a twice continuously differentiable convex function, $p:\mathbb{R}^m\to(-\infty,+\infty]$ is a closed proper convex function, and $B:\mathbb{R}^n\to\mathbb{R}^m$ is a linear map. In many real applications, p is a regularizer which can enforce sparsity, and the linear map B encodes desirable structures of x. This indicates the meaning of structured sparsity. The optimization problem (1.2) includes many important models, such as the convex clustering model (1.1),[1] fused lasso model [34], clustered lasso model [27], and so on.

When the linear map B in (1.2) is an identity map, various feature screening rules have been proposed to reduce the computational cost for generating the solution path. The feature screening rules attempt to exploit the sparsity induced by the regularization function p on the variable x and to drop some inactive features based on prior analysis before we solve the optimization problem [3, 7, 8, 14, 33, 36, 37]. More recently, Lin et al. [16] proposed an adaptive sieving (AS) technique to reduce the dimension of the optimization problem with sparse solutions for a general regularization function $p(\cdot)$. Here, we briefly compare the AS technique and the fast reduced space algorithm (FaRSA) for the lasso problem [3] and the nonoverlapping group lasso problem [7]. The feature screening ideas behind the AS technique and the FaRSA share some similarities. However, the FaRSA is more akin to the active set method [14], and the reduced subproblems are constructed based on a quadratic approximation of the smooth function $f(\cdot)$. Instead, the AS technique constructs the reduced subproblem without any approximation. Readers can refer to [16] and the references therein for more details about the AS technique and its comparison with other feature screening rules. But note that all the existing feature screening rules and dimension reduction techniques are not applicable to (1.2) with structured sparsity, when the solution x itself may not be sparse.

[1] We can take $x=\mathrm{vec}(X)\in\mathbb{R}^{dN}$, where $\mathrm{vec}(X)$ is the vectorization of the matrix X by stacking its columns one by one.

In this paper, we will generalize the ideas in [16] to propose new dimension reduction techniques, which can be used to solve the optimization problem (1.2) via solving a sequence of subproblems with much smaller problem size. Here, we briefly discuss the challenges to generalize the AS technique to (1.2). First, the optimal solutions of (1.2) may be dense, only with some special structures, such as being blockwise constant. Thus, the AS technique in [16] could not be applied to x. Second, one may try to apply AS directly to Bx by introducing a new variable $y = Bx$. Although this idea may work for reducing the dimension of y (or Bx), it cannot reduce the dimension of x simultaneously. As a result, we still need to solve large-scale subproblems. Third, in order to apply the AS technique, we need to check the optimality condition of (1.2) for a given $\bar{x} \in \mathbb{R}^n$. However, as one may see later, this is highly nontrivial if the inverse of B is not available. In this paper, we will propose a new AS technique and an enhanced AS (EAS) technique to tackle these issues.

To demonstrate the effectiveness of the proposed idea, we evaluate the empirical performance of the AS and EAS techniques with the state-of-the-art algorithms Ssnal (a semismooth Newton-CG augmented Lagrangian method) [39], ADMM (alternating direction method of multipliers) [4], and AMA (alternating minimization algorithm) [4] for solving the convex clustering model (1.1). As the readers will see later, the numerical results on both simulated and real datasets demonstrate that the proposed AS and EAS techniques could substantially reduce the dimension of the optimization problems. As a result, the AS and EAS techniques can accelerate the state-of-the-art algorithm Ssnal by more than 7 times and the algorithm ADMM by more than 14 times for solving the convex clustering model. To further demonstrate the generality of the AS and EAS techniques, we conduct additional numerical experiments on the overlapping group lasso model [11]; the details can be found in Appendix A.

We summarize our main contributions of this paper as follows:

- We propose a new AS technique which can solve large-scale optimization problems of the form (1.2) with structured sparsity by solving a sequence of reduced subproblems with smaller sizes.
- The proposed AS technique can reduce the dimension of x and Bx simultaneously. We show how to construct the corresponding reduced subproblem of (1.2) based on the structured sparsity of x (i.e., the sparsity of Bx).
- The proposed AS technique allows the reduced subproblems to be solved inexactly. We prove the finite convergence property of the proposed AS technique for obtaining an ϵ-optimal solution to (1.2) for a given tolerance $\epsilon > 0$.
- Although the AS technique will converge in finite iterations, the sieving procedure of the AS technique may continue even if we only want to obtain an ϵ-optimal solution of (1.2). To address this issue, we propose an EAS technique, which can certify the ϵ-optimality of an obtained solution with affordable computation cost. This can potentially reduce the sieving iterations and further accelerate the algorithms. The finite convergence of the EAS technique is also proved.
- Both the AS and EAS techniques are extended to obtain a solution path of the structured sparse optimization problem (1.2) for a sequence of hyperparameters $+\infty > \lambda_1 > \lambda_2 > \cdots > \lambda_k > 0$.
- As an application, extensive numerical experiments on the convex clustering model for both simulated and real datasets are provided. The superior numerical experiment results demonstrate the power of the AS and EAS techniques for accelerating numerical optimization algorithms to generate the solution path for the convex clustering model (1.1).

The rest of this paper is organized as follows. In section 2, we introduce the AS technique and the EAS technique for convex optimization problems with structured sparsity. The application of the AS and EAS techniques on the convex clustering model will be shown in section 3. Numerical results on the convex clustering are presented in section 4. We conclude the paper in section 5. Additional numerical results on the overlapping group lasso model [11] are presented in Appendix A.

Notation. We use $\mathbb{R}^{m\times n}$ ($\mathbb{R}^n$) to denote the set of all real $m\times n$ matrices (n-dimensional vectors). We denote column vectors by lowercase letters, e.g., $v\in\mathbb{R}^n$, and matrices by capital letters, e.g., $A\in\mathbb{R}^{m\times n}$. We denote the transpose of the matrix A as A^T and the ith (ijth) element of a vector v (matrix A) by v_i (A_{ij}). For a given integer $n\geq 1$, we denote the collection of integers from 1 to n by $[n]$. We denote the complement of an index set $I\subseteq[m]$ as I^c. For given index sets $I\subseteq[m]$ and $J\subseteq[n]$, we denote the submatrix consisting with rows (columns) indexed by I (J) as $A_{I:}$ ($A_{:J}$). We denote the range space and null space of A by Range(A) and Null(A), respectively. For a vector $x\in\mathbb{R}^n$ and a scalar $p>0$, we define the vector p-norm as $\|x\|_p:=(\sum_{i=1}^n|x_i|^p)^{1/p}$. We use $\|\cdot\|$ to denote the vector 2-norm. For a closed proper convex function $f:\mathbb{R}^n\to(-\infty,+\infty]$, the conjugate of f is $f^*(z):=\sup_{x\in\mathbb{R}^n}\{\langle x,z\rangle-f(x)\}$. In addition, we define the proximal mapping of f as $\mathrm{Prox}_f(x):=\arg\min_{z\in\mathbb{R}^n}f(z)+\frac{1}{2}\|z-x\|^2$. For a closed convex set $C\subseteq\mathbb{R}^n$ and a given vector $a\in\mathbb{R}^n$, the projection of a onto the set C is $\Pi_C(a):=\arg\min_{x\in C}\frac{1}{2}\|x-a\|^2$.

2. An AS technique for structured sparsity. In this section, we will introduce a novel AS technique for obtaining the solution path for the structured sparse convex programming problem (1.2). Equivalently, we can reformulate (1.2) as follows:

$$(P_\lambda)\qquad \begin{array}{ll}\min_{x\in\mathbb{R}^n,y\in\mathbb{R}^m} & f(x)+\lambda p(y)\\ \text{s.t.} & Bx-y=0.\end{array}$$

Throughout this paper, we assume that the following constraint qualification for (P_λ) holds:

$$\text{(CQ)}\qquad 0\in\mathrm{ri}(\mathrm{dom}(p)),$$

where $\mathrm{ri}(\mathrm{dom}(p))$ is the relative interior of $\mathrm{dom}(p)$, the domain of p. The Lagrangian function corresponding to (P_λ) is defined as

$$l(x,y;z):=f(x)+\lambda p(y)+\langle z,Bx-y\rangle, \tag{2.1}$$

where $z\in\mathbb{R}^m$ is the Lagrange multiplier. The corresponding dual problem is

$$(D_\lambda)\qquad \max_{z\in\mathbb{R}^m}\quad D_\lambda(z):=-f^*(-B^Tz)-\lambda p^*(z/\lambda).$$

Here, f^* and p^* are the conjugate of f and p, respectively. Let Ω_λ be the optimal solution set of(P_λ). Under (CQ), it follows from [26, Corollaries 28.2.2 and 28.3.1] that $(x^*,y^*)\in\Omega_\lambda$ if and only if there exists $z^*\in\mathbb{R}^m$ such that

$$\text{(KKT)}\qquad \begin{cases}\nabla f(x^*)+B^Tz^*=0,\\ z^*\in\lambda\partial p(y^*),\\ Bx^*-y^*=0.\end{cases}$$

Thus, we can define the KKT residual function for problem (P_λ) as

$$R_\lambda(x,y,z):=\begin{pmatrix}\nabla f(x)+B^Tz\\ y-\mathrm{Prox}_{\lambda p}(y+z)\\ Bx-y\end{pmatrix}\quad\forall\,(x,y,z)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^m. \tag{2.2}$$

We know that $(x^*, y^*) \in \Omega_\lambda$ if and only if there exists $z^* \in \mathbb{R}^m$ such that

$$R_\lambda(x^*, y^*, z^*) = 0.$$

For a given $\epsilon > 0$, we call $(\bar{x}, \bar{y})$ an ϵ-optimal solution to (P_λ) if there exists $\bar{z} \in \mathbb{R}^m$ such that $\|R_\lambda(\bar{x}, \bar{y}, \bar{z})\| \le \epsilon$. We also call such a $(\bar{x}, \bar{y}, \bar{z})$ an ϵ-KKT point to (P_λ).

In this paper, we make the following two mild assumptions.

Assumption 2.1. For any given $\lambda > 0$, the optimal solution set Ω_λ to the optimization problem (P_λ) is nonempty and compact.

Assumption 2.2. For any given $\lambda > 0$ and $y \in \mathbb{R}^m$, if $I_0^c \neq \emptyset$, then $(\partial(\lambda p(y)))_{I_0^c}$ is a singleton, where

$$I_0^c := \{i \in [m] \mid y_i \neq 0\}.$$

Remark 2.1. We make some remarks on Assumption 2.2. For most of the commonly used regularizers, such as lasso [32], group lasso [38], and exclusive lasso [41], Assumption 2.2 is satisfied. Let us take the lasso regularizer as an example. If $p(y) = \|y\|_1$, then

$$(\partial(\lambda p(y)))_i = \begin{cases} \lambda \mathrm{sign}(y_i) & \text{if } y_i \neq 0, \\ [-\lambda, \lambda] & \text{if } y_i = 0. \end{cases}$$

Thus, we know that for any $y \neq 0$, $(\partial(\lambda p(y)))_{I_0^c} = (\lambda \mathrm{sign}(y))_{I_0^c}$ is a singleton. Here $\mathrm{sign}(\cdot)$ is the signum function.

The theme of this paper is to design a technique which can reduce the dimension of a class of optimization problems of the form (P_λ) with structured sparsity by exploring the intrinsic structure of the problem in an explicit way.

We first introduce our principal idea in a general way. Then, we will propose a new AS technique to rigorously implement the idea. We fix the parameter λ in (P_λ) for now. For a given index set $I \subseteq [m]$, if there is some prior knowledge for us to assume that $y_I = 0$, then it is natural to consider the following constrained optimization problem generated by the index set I:

$$(P_\lambda(I)) \qquad \begin{array}{ll} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} & f(x) + \lambda p(y) \\ \text{s.t.} & Bx - y = 0, \\ & y_I = 0. \end{array}$$

We denote this problem as $(P_\lambda(I))$ to indicate its dependence on the index set I.

Our principal idea is to obtain a solution to the original optimization problem (P_λ) by solving a sequence of subproblems with lower dimension, which are induced by $(P_\lambda(I))$. The keys to achieve this goal depend on answering the following questions:

- Q1: For a given index set $I \subseteq [m]$, how can we effectively reduce the dimension of (P_λ) and construct a proper reduced problem $(RP_\lambda(I))$ (the exact form of $(RP_\lambda(I))$ is given in section 2.1)?
- Q2: If the current obtained solution pair $(\bar{x}, \bar{y})$ of $(P_\lambda(I))$ is *not* yet an ϵ-optimal solution to (P_λ), how can we update the index set I to construct a new problem in the form of $(P_\lambda(I))$?
- Q3: If an obtained solution pair $(\bar{x}, \bar{y})$ of $(P_\lambda(I))$ is indeed an ϵ-optimal solution to (P_λ), can we certify this and stop the whole procedure?
- Q4: Is the proposed technique robust to the inexactness of the obtained solution pair? In other words, if we can only obtain an inexact solution of $(RP_\lambda(I))$ under a given tolerance $\epsilon > 0$, can we obtain an inexact solution of (P_λ) under the tolerance $O(\epsilon)$?

Q5: Is it possible to apply the proposed technique to any algorithms that can obtain an ϵ-KKT point of $(RP_\lambda(I))$ under a given tolerance $\epsilon > 0$?

Remark 2.2. We make some remarks before we describe the AS technique.

1. Although designing an efficient and convergent algorithm for solving $(P_\lambda(I))$ is also an important task, it is not the main purpose of this paper. There are also existing algorithms which can solve $(RP_\lambda(I))$ and obtain an ϵ-KKT point with a moderate accuracy [4, 30, 39].
2. Question Q3 is essential for applying a dimension reduction technique to solving (P_λ) based on $(P_\lambda(I))$. In order to check the optimality (or ϵ-optimality) of the solution pair $(\bar{x}, \bar{y})$, we need to construct the corresponding dual solution and check the KKT conditions. This is highly nontrivial since the dual solutions are not unique if structured sparsity exists. This is also one of the main difficulties for applying the AS technique to problem (P_λ) with structured sparsity as compared to [16].
3. The robustness mentioned in Q4 is necessary, since the best we can expect in general is to obtain an inexact solution to $(RP_\lambda(I))$.

2.1. A dimension reduction technique for (P_λ) based on $(P_\lambda(I))$. We first show how we can reduce the dimension of the variables x and y simultaneously for the problem (P_λ) based on the constrained optimization problem $(P_\lambda(I))$. This will answer the question Q1.

Assume that the rank of $B_{I:}$ is $r > 0$. Then there exist three index sets α, β, and γ with $|\gamma| = r$ that form a partition of $[n]$ such that $B_{I\beta} = 0$ and $B_{I\gamma}$ has full column rank. Here, we assume that the index set α is nonempty; otherwise, we must have $x_\gamma = 0$.

Since $B_{I\gamma}$ has full column rank, there is a unique $|\gamma| \times |\alpha|$ matrix $M_{\gamma\alpha}$ such that[2]

$$B_{I\alpha} + B_{I\gamma} M_{\gamma\alpha} = 0. \tag{2.3}$$

Then, we can eliminate x_γ by $y_I = 0$ of $(P_\lambda(I))$ as

$$x_\gamma = M_{\gamma\alpha} x_\alpha.$$

Define

$$\varphi(x_\alpha, x_\beta) = f(\dot{x}), \quad q(y_{I^c}) = p(\dot{y}),$$

where

$$\dot{x}_\alpha = x_\alpha, \quad \dot{x}_\beta = x_\beta, \quad \dot{x}_\gamma = M_{\gamma\alpha} x_\alpha,$$

and

$$\dot{y}_i = \begin{cases} y_i & \text{if } i \in I^c, \\ 0 & \text{if } i \in I. \end{cases}$$

Then, we can obtain a solution to $(P_\lambda(I))$ via solving the following reduced optimization problem:

$$(RP_\lambda(I)) \quad \begin{array}{ll} \min_{x_\alpha \in \mathbb{R}^{|\alpha|}, x_\beta \in \mathbb{R}^{|\beta|}, y_{I^c} \in \mathbb{R}^{|I^c|}} & \varphi(x_\alpha, x_\beta) + \lambda q(y_{I^c}) \\ \text{s.t.} & (B_{I^c\alpha} + B_{I^c\gamma} M_{\gamma\alpha}) x_\alpha + B_{I^c\beta} x_\beta - y_{I^c} = 0. \end{array}$$

The Lagrange function corresponding to $(RP_\lambda(I))$ is given by

$$l(x_\alpha, x_\beta, y_{I^c}, \xi) = \varphi(x_\alpha, x_\beta) + \lambda q(y_{I^c}) + \langle \xi, (B_{I^c\alpha} + B_{I^c\gamma} M_{\gamma\alpha}) x_\alpha + B_{I^c\beta} x_\beta - y_{I^c} \rangle,$$

where $\xi \in \mathbb{R}^{|I^c|}$ is the Lagrange multiplier.

[2]Here, we slightly abuse the notation to indicate the dependence of $M_{\gamma\alpha}$ on the index sets α and γ. The uniqueness is in the sense of a given partition.

Now, if we solve $(RP_\lambda(I))$ to obtain a solution $(\hat{x}_\alpha, \hat{x}_\beta, \hat{y}_{I^c})$, then there exists a $\hat{\xi}$ that satisfies the following KKT conditions:

$$\begin{cases} (\nabla f(\hat{x}))_\alpha + M_{\gamma\alpha}^T(\nabla f(\hat{x}))_\gamma + (B_{I^c\alpha} + B_{I^c\gamma}M_{\gamma\alpha})^T\hat{\xi} = 0, \\ (\nabla f(\hat{x}))_\beta + B_{I^c\beta}^T\hat{\xi} = 0, \quad \hat{\xi} \in (\partial(\lambda p(\hat{y})))_{I^c}, \\ (B_{I^c\alpha} + B_{I^c\gamma}M_{\gamma\alpha})\hat{x}_\alpha + B_{I^c\beta}\hat{x}_\beta - \hat{y}_{I^c} = 0, \end{cases} \tag{2.4}$$

where $\hat{x}$ and $\hat{y}$ are defined as

$$\hat{x}_\alpha = \hat{x}_\alpha, \quad \hat{x}_\beta = \hat{x}_\beta, \quad \hat{x}_\gamma = M_{\gamma\alpha}\hat{x}_\alpha$$

and

$$\hat{y}_{I^c} = \hat{y}_{I^c}, \quad \hat{y}_I = 0,$$

respectively. Then $(\bar{x}, \bar{y})$, which is constructed by

$$\begin{cases} \bar{x}_\alpha = \hat{x}_\alpha, \quad \bar{x}_\beta = \hat{x}_\beta, \quad \bar{x}_\gamma = M_{\gamma\alpha}\hat{x}_\alpha, \\ \bar{y}_{I^c} = \hat{y}_{I^c}, \quad \bar{y}_I = 0, \end{cases} \tag{2.5}$$

is a solution to problem $(P_\lambda(I))$. Thus, in order to obtain a solution to $(P_\lambda(I))$, we only need to solve a corresponding reduced problem $(RP_\lambda(I))$ whose dimension can be much smaller.

Remark 2.3. We make some remarks to close this subsection.

1. We reduce the dimension of the problem from $\mathbb{R}^n \times \mathbb{R}^m$ to $\mathbb{R}^{n-|\gamma|} \times \mathbb{R}^{m-|I|}$, which can be a substantial reduction. For example, if the solution of (P_λ) is indeed sparse (this is an intrinsic property since we can obtain a sparse solution in general for large λ), then $|I|$ is close to m and $|\gamma|$ is close to n simultaneously.
2. In general, it is nontrivial to identify α, β, and γ and to construct the matrix $M_{\gamma\alpha}$. However, in many real applications, the construction can be done at a low cost due to the special structure of the matrix B. Some examples can be found in section 3 and Appendix A, respectively.

2.2. An AS technique for (P_λ) with a fixed $\lambda > 0$. We move on to present the details of the AS technique. We fix the parameter λ for now, and we will generalize it to handle the case for a sequence of $\lambda > 0$ later. Also, for simplicity, we first present the idea with the assumption that we can solve $(RP_\lambda(I))$ exactly. The same idea will be generalized to the inexact setting without much difficulties later.

We first show how we can update the index set I if the current obtained solution $(\bar{x}, \bar{y})$ via solving $(P_\lambda(I))$ is not an optimal solution to (P_λ). The key idea is to construct a corresponding dual variable pair $(\bar{u}, \bar{w}) \in \mathbb{R}^m \times \mathbb{R}^{|I|}$ which satisfies the following KKT conditions for $(P_\lambda(I))$:

$$\begin{cases} (\nabla f(\bar{x}))_\alpha + B_{I\alpha}^T\bar{u}_I + B_{I^c\alpha}^T\bar{u}_{I^c} = 0, \\ (\nabla f(\bar{x}))_\beta + B_{I^c\beta}^T\bar{u}_{I^c} = 0, \quad (\nabla f(\bar{x}))_\gamma + B_{I\gamma}^T\bar{u}_I + B_{I^c\gamma}^T\bar{u}_{I^c} = 0, \\ \bar{u}_I - \bar{w} \in \lambda(\partial p(\bar{y}))_I, \quad \bar{u}_{I^c} \in \lambda(\partial p(\bar{y}))_{I^c}, \\ B\bar{x} - \bar{y} = 0, \quad \bar{y}_I = 0. \end{cases} \tag{2.6}$$

Since $(\bar{x}, \bar{y}) = (\hat{x}, \hat{y})$ and $(\hat{x}, \hat{y}, \hat{\xi})$ is a solution to (2.4), we must have

$$B_{I^c\beta}^T\hat{\xi} = B_{I^c\beta}^T\bar{u}_{I^c}, \quad \hat{\xi} \in (\partial(\lambda p(\hat{y})))_{I^c}, \quad \bar{u}_{I^c} \in (\partial(\lambda p(\bar{y})))_{I^c}.$$

Aggressively, we construct $\bar{u}_{I^c}$ as

$$\bar{u}_{I^c} = \hat{\xi}. \tag{2.7}$$

By the above construction of $\bar{u}_{I^c}$ and (2.3), the first equation of (2.6) is implied by the third equation of (2.6) and the first equation of (2.4). Thus, we can construct the pair $(\bar{u}_I, \bar{w})$ via solving the following equations for (u_I, w):

$$\begin{cases} (\nabla f(\bar{x}))_\gamma + B_{I\gamma}^T u_I + B_{I^c\gamma}^T \bar{u}_{I^c} = 0, \\ u_I - w \in (\partial(\lambda p(\bar{y})))_I. \end{cases} \tag{2.8}$$

Since w is an unconstrained variable, for any $\hat{u}_I$ satisfying the first equation of (2.8), there exists a $\hat{w}$ such that the inclusion in (2.8) is satisfied. Then, consider the fact that if there exists a $\tilde{u}_I$ such that $(\tilde{u}_I, 0)$ is a solution to (2.8), then the current solution pair $(\bar{x}, \bar{y})$ is an optimal solution to (P_λ). Motivated by this fact, we propose to construct the pair $(\bar{u}_I, \bar{w})$ such that $\bar{w}$ achieves the minimum Euclidean norm. Since $B_{I\gamma}$ has full column rank, we can construct a particular solution to the first equation of (2.8) as

$$(\bar{u}_I)_0 = -B_{I\gamma}(B_{I\gamma}^T B_{I\gamma})^{-1}((\nabla f(\bar{x}))_\gamma + B_{I^c\gamma}^T \bar{u}_{I^c}). \tag{2.9}$$

Thus, all the solutions to the first equation of (2.8) are given by

$$u_I = (\bar{u}_I)_0 + d,$$

where $d \in \mathrm{Null}(B_{I\gamma}^T)$. In summary, we construct the solution pair $(\bar{u}_I, \bar{w})$ as follows:

$$\bar{u}_I = (\bar{u}_I)_0 + \bar{d}, \quad \bar{w} = \bar{u}_I - \Pi_{(\partial(\lambda p(\bar{y})))_I}(\bar{u}_I), \tag{2.10}$$

where $\bar{d}$ is a solution to the following auxiliary optimization problem:

$$\begin{array}{ll} \min_{d \in \mathbb{R}^{|I|}} & \frac{1}{2}\|((\bar{u}_I)_0 + d) - \Pi_{(\partial(\lambda p(\bar{y})))_I}((\bar{u}_I)_0 + d)\|^2 \\ \text{s.t.} & d \in \mathrm{Null}(B_{I\gamma}^T). \end{array} \tag{2.11}$$

Up to this point, we have completed the construction of a dual solution pair $(\bar{u}, \bar{w})$. We show the properties of the constructed $(\bar{u}, \bar{w})$ in Theorem 2.4 and Theorem 2.6.

THEOREM 2.4. *Assume that $(\hat{x}_\alpha, \hat{x}_\beta, \hat{y}_{I^c})$ is an optimal solution to the optimization problem*

(2.12)
$$\begin{array}{ll} \min_{x_\alpha \in \mathbb{R}^{|\alpha|}, x_\beta \in \mathbb{R}^{|\beta|}, y_{I^c} \in \mathbb{R}^{|I^c|}} & \varphi(x_\alpha, x_\beta) + \lambda q(y_{I^c}) + \langle x_\alpha, \hat{\delta}_1 \rangle + \langle x_\beta, \hat{\delta}_2 \rangle - \langle y_{I^c}, \hat{\delta}_3 \rangle \\ \text{s.t.} & (B_{I^c\alpha} + B_{I^c\gamma} M_{\gamma\alpha}) x_\alpha + B_{I^c\beta} x_\beta - y_{I^c} = 0 \end{array}$$

and $\hat{\xi}$ is the corresponding Lagrange multiplier. Here, $(\hat{\delta}_1, \hat{\delta}_2, \hat{\delta}_3) \in \mathbb{R}^{|\alpha|} \times \mathbb{R}^{|\beta|} \times \mathbb{R}^{|I^c|}$ are given error terms satisfying $\|\hat{\delta}_1\| + \|\hat{\delta}_2\| + \|\hat{\delta}_3\| \le \epsilon$. Then, $(\hat{x}_\alpha, \hat{x}_\beta, \hat{y}_{I^c})$ is an ϵ-optimal solution to $(RP_\lambda(I))$. Let $(\bar{x}, \bar{y}, \bar{u}_{I^c}, \bar{u}_I, \bar{w})$ be the solution that is constructed from (2.5), (2.7), *and* (2.10). *Define $J(\lambda)$ as follows:*

$$J(\lambda) := \{j \in I \mid \bar{u}_j \notin (\partial(\lambda p(\bar{y})))_j\}. \tag{2.13}$$

Then, $J(\lambda) \neq \emptyset$ if

$$\|R_\lambda(\bar{x}, \bar{y}, \bar{u})\| > \epsilon.$$

Proof. Since $(\hat{x}_\alpha, \hat{x}_\beta, \hat{y}_{I^c})$ is an optimal solution to (2.12) and $\hat{\xi}$ is the corresponding Lagrange multiplier, the following KKT system holds:

$$\begin{cases} (\nabla f(\hat{x}))_\alpha + M_{\gamma\alpha}^T(\nabla f(\hat{x}))_\gamma + (B_{I^c\alpha} + B_{I^c\gamma}M_{\gamma\alpha})^T\hat{\xi} + \hat{\delta}_1 = 0, \\ (\nabla f(\hat{x}))_\beta + B_{I^c\beta}^T\hat{\xi} + \hat{\delta}_2 = 0, \\ \hat{\xi} + \hat{\delta}_3 \in (\partial(\lambda p(\hat{y})))_{I^c}, \\ (B_{I^c\alpha} + B_{I^c\gamma}M_{\gamma\alpha})\hat{x}_\alpha + B_{I^c\beta}\hat{x}_\beta - \hat{y}_{I^c} = 0. \end{cases} \tag{2.14}$$

Since $\|\hat{\delta}_1\| + \|\hat{\delta}_2\| + \|\hat{\delta}_3\| \le \epsilon$, by (2.4), (2.14), and the property that the proximal mapping is Lipschitz continuous with modulus 1, we can verify that $(\hat{x}_\alpha, \hat{x}_\beta, \hat{y}_{I^c})$ is an ϵ-optimal solution to $(RP_\lambda(I))$.

By construction, $(\bar{x}, \bar{y}, \bar{u}_{I^c}, \bar{u}_I, \bar{w})$ is a solution to

$$\begin{cases} (\nabla f(\bar{x}))_\alpha + B_{I\alpha}^T\bar{u}_I + B_{I^c\alpha}^T\bar{u}_{I^c} + \hat{\delta}_1 = 0, \\ (\nabla f(\bar{x}))_\beta + B_{I^c\beta}^T\bar{u}_{I^c} + \hat{\delta}_2 = 0, \\ (\nabla f(\bar{x}))_\gamma + B_{I\gamma}^T\bar{u}_I + B_{I^c\gamma}^T\bar{u}_{I^c} = 0, \\ \bar{u}_I - \bar{w} \in \lambda(\partial p(\bar{y}))_I, \\ \bar{u}_{I^c} + \hat{\delta}_3 \in \lambda(\partial p(\bar{y}))_{I^c}, \\ B\bar{x} - \bar{y} = 0, \quad \bar{y}_I = 0. \end{cases} \tag{2.15}$$

Now, we prove that $J(\lambda) \neq \emptyset$ provided $\|R_\lambda(\bar{x}, \bar{y}, \bar{u})\| > \epsilon$. We prove it by contradiction. Assume that

$$J(\lambda) = \emptyset.$$

Then we have

$$\bar{u} + \hat{\delta} \in \partial(\lambda p(\bar{y})),$$

where $\hat{\delta} = (\hat{\delta}_I, \hat{\delta}_{I^c}) = (0, \hat{\delta}_3)$. This implies that

$$\bar{y} - \mathrm{Prox}_{\lambda p}(\bar{y} + (\bar{u} + \hat{\delta})) = 0.$$

Then,

$$\begin{aligned} \|R_\lambda(\bar{x}, \bar{y}, \bar{u})\| &= \|(\nabla f(\bar{x}) + B^T\bar{u}, \bar{y} - \mathrm{Prox}_{\lambda p}(\bar{y} + \bar{u}), B\bar{x} - \bar{y})\| \\ &= \|((-\hat{\delta}_1, -\hat{\delta}_2, 0), \mathrm{Prox}_{\lambda p}(\bar{y} + (\bar{u} + \hat{\delta})) - \mathrm{Prox}_{\lambda p}(\bar{y} + \bar{u}), 0)\| \\ &\le \|\hat{\delta}_1\| + \|\hat{\delta}_2\| + \|\hat{\delta}_3\| \\ &\le \epsilon. \end{aligned}$$

Here, we use the property that the proximal mapping is Lipschitz continuous with modulus 1. This is a contradiction. Thus $J(\lambda) \neq \emptyset$. ∎

Remark 2.5. We do not need to specify a priori error terms $\hat{\delta}_1, \hat{\delta}_2, \hat{\delta}_3$ in Theorem 2.4. They should be interpreted as the errors incurred when we solve the problem $(RP_\lambda(I))$ inexactly with a given tolerance.

An implication of Theorem 2.4 is that, if the current obtained solution pair $(\bar{x}, \bar{y})$ is not an ϵ-optimal solution to (P_λ) under the given tolerance, we can update the index set I by removing the identified violated index set $J(\lambda)$. This motivates us to propose the AS technique for (P_λ) with a given fixed $\lambda > 0$, which is presented in Algorithm 2.1.

Algorithm 2.1 AS for solving (P_λ) with a fixed $\lambda > 0$

Input: a given hyperparameter $\lambda > 0$ and a given tolerance $\epsilon > 0$.
Output: $(x^*(\lambda), y^*(\lambda), z^*(\lambda))$.
Initialization: Generate an initial index set by a predefined initialization strategy: $I^0(\lambda) \subseteq [m]$.
for $i = 0, 1, 2, \ldots$ **do**
  1. For the given index set $I^i(\lambda)$, construct the index partition $\{\alpha^i, \beta^i, \gamma^i\}$ and the corresponding $M_{\gamma^i\alpha^i}$.
  2. Apply any well designed algorithm to solving problem $(RP_\lambda(I))$ with the constructed $\{I^i(\lambda), \alpha^i, \beta^i, \gamma^i, M_{\gamma^i\alpha^i}\}$, and obtain an inexact solution $(\hat{x}^i_{\alpha_i}, \hat{x}^i_{\beta_i}, \hat{y}^i_{(I^i)^c(\lambda)}, \hat{\xi}^i)$ which satisfies the corresponding KKT system (2.14) with the latent error terms $(\hat{\delta}^i_1, \hat{\delta}^i_2, \hat{\delta}^i_3)$ such that $\|\hat{\delta}^i_1\| + \|\hat{\delta}^i_2\| + \|\hat{\delta}^i_3\| \le \epsilon$.
  3. Recover a solution $(\bar{x}^i, \bar{y}^i, \bar{u}^i, \bar{w}^i)$ by the construction of (2.5), (2.7), and (2.10), respectively.
  if $\|R_\lambda(\bar{x}^i, \bar{y}^i, \bar{u}^i)\| \le \epsilon$ **then**
    Set $(x^*(\lambda), y^*(\lambda), z^*(\lambda)) = (\bar{x}^i, \bar{y}^i, \bar{u}^i)$.
    break.
  else
    Create $J^i(\lambda)$:

$$J^i(\lambda) = \{j \in I^i(\lambda) \mid \bar{u}^i_j \notin \partial(\lambda p(\bar{y}^i))_j\}. \tag{2.16}$$

    Update $I^{i+1}(\lambda)$ as

$$I^{i+1}(\lambda) \leftarrow I^i(\lambda) \backslash J^i(\lambda).$$

  end if
end for
return $(x^*(\lambda), y^*(\lambda), z^*(\lambda))$.

THEOREM 2.6. *For a given $\epsilon > 0$, with any well designed algorithm which can solve the reduced subproblem $(RP_\lambda(I))$ to the given accuracy, Algorithm* 2.1 *is guaranteed to converge in a finite number of iterations. Moreover, the obtained pair $(x^*(\lambda), y^*(\lambda), z^*(\lambda))$ is a solution to (P_λ) in the sense that*

$$\|R_\lambda(x^*(\lambda), y^*(\lambda), z^*(\lambda))\| \le \epsilon.$$

We omit the proof of Theorem 2.6 as it is a byproduct of Theorem 2.4.

Remark 2.7. We close this subsection by making some remarks here.

1. The proposed AS technique is an implementation of the aforementioned principal idea, which simultaneously answers the questions Q1, Q2, Q4, and Q5.
2. The AS technique can be applied to any algorithm which can obtain an ϵ-KKT point to $(RP_\lambda(I))$. In particular, classical algorithms such as the augmented Lagrangian method and ADMM are applicable.
3. However, it may fail to answer the question Q3. The whole procedure described in Algorithm 2.1 is not guaranteed to certify the ϵ-optimality of a given solution pair $(\bar{x}, \bar{y})$ for (P_λ). The main reason is that we have aggressively set $\bar{u}_{I^c} = \hat{\xi}$ in (2.7).
4. Although Algorithm 2.1 may fail to answer the question Q3 and it may need additional sieving iterations, the practical performance of Algorithm 2.1 is promising. Readers can find the numerical performance in section 4 and Appendix A.

2.3. An EAS technique. Now, we introduce an EAS technique, which can certify the ϵ-optimality of the obtained pair $(\bar{x}, \bar{y})$ via solving $(RP_\lambda(I))$ if it is an ϵ-optimal solution to (P_λ). Thus, we can potentially reduce the number of sieving iterations of Algorithm 2.1.

The key idea is to deal with the issues we mentioned in Remark 2.7. Now, assume that $(\bar{x}, \bar{y})$ is an optimal solution to $(P_\lambda(I))$, which could be recovered by (2.5) with a solution of $(RP_\lambda(I))$. We can then define a new index set $\tilde{I}$ as follows:

$$\tilde{I} := \{i \in [m] \mid \bar{y}_i = 0\}. \tag{2.17}$$

By the construction, we have $I \subseteq \tilde{I}$. It is not difficult to see that $(\bar{x}, \bar{y})$ is an optimal solution to the following constrained optimization problem:

$$(P_\lambda(\tilde{I})) \qquad \begin{array}{ll} \min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} & f(x) + \lambda p(y) \\ \text{s.t.} & Bx - y = 0, \\ & y_{\tilde{I}} = 0. \end{array}$$

In a similar manner, we can define the index sets $\tilde{\alpha}$, $\tilde{\beta}$, and $\tilde{\gamma}$ with $|\tilde{\gamma}| = \tilde{r}$, which form a partition of $[n]$, such that $B_{\tilde{I}\tilde{\beta}} = 0$ and $B_{\tilde{I}\tilde{\gamma}}$ has full column rank. Again, we assume that $\tilde{\alpha} \neq \emptyset$. Thus, there exists an $M_{\tilde{\gamma}\tilde{\alpha}} \in \mathbb{R}^{|\tilde{\gamma}|\times|\tilde{\alpha}|}$ such that

$$B_{\tilde{I}\tilde{\alpha}} + B_{\tilde{I}\tilde{\gamma}} M_{\tilde{\gamma}\tilde{\alpha}} = 0.$$

Then, we can eliminate $x_{\tilde{\gamma}}$ by the constraints of $(P_\lambda(\tilde{I}))$ as

$$x_{\tilde{\gamma}} = M_{\tilde{\gamma}\tilde{\alpha}} x_{\tilde{\alpha}}.$$

The Lagrangian function corresponding to $(P_\lambda(\tilde{I}))$ is

$$l(x, y, v, s) = f(x) + \lambda p(y) + \langle v, Bx - y\rangle + \langle s, y_{\tilde{I}}\rangle,$$

where $v \in \mathbb{R}^m$ and $s \in \mathbb{R}^{|\tilde{I}|}$ are the Lagrange multipliers. For notational consistency, we denote $(\tilde{x}, \tilde{y}) = (\bar{x}, \bar{y})$. Since $(\tilde{x}, \tilde{y})$ is an optimal solution to $(P_\lambda(\tilde{I}))$, there exists $(\tilde{v}, \tilde{s})$ satisfying the following KKT conditions for $(P_\lambda(\tilde{I}))$:

$$\begin{cases} (\nabla f(\tilde{x}))_{\tilde{\alpha}} + B^T_{\tilde{I}\tilde{\alpha}}\tilde{v}_{\tilde{I}} + B^T_{\tilde{I}^c\tilde{\alpha}}\tilde{v}_{\tilde{I}^c} = 0, \quad (\nabla f(\tilde{x}))_{\tilde{\beta}} + B^T_{\tilde{I}^c\tilde{\beta}}\tilde{v}_{\tilde{I}^c} = 0, \\ (\nabla f(\tilde{x}))_{\tilde{\gamma}} + B^T_{\tilde{I}\tilde{\gamma}}\tilde{v}_{\tilde{I}} + B^T_{\tilde{I}^c\tilde{\gamma}}\tilde{v}_{\tilde{I}^c} = 0, \\ \tilde{v}_{\tilde{I}} - \tilde{s} \in \lambda(\partial p(\tilde{y}))_{\tilde{I}}, \quad \tilde{v}_{\tilde{I}^c} \in \lambda(\partial p(\tilde{y}))_{\tilde{I}^c}, \\ B\tilde{x} - \tilde{y} = 0, \quad \tilde{y}_{\tilde{I}} = 0. \end{cases} \tag{2.18}$$

On the other hand, we know that $(\tilde{x}_{\tilde{\alpha}}, \tilde{x}_{\tilde{\beta}}, \tilde{y}_{\tilde{I}^c})$ is an optimal solution to the following reduced problem corresponding to $(P_\lambda(\tilde{I}))$:

$$(RP_\lambda(\tilde{I})) \qquad \begin{array}{ll} \min_{x_{\tilde{\alpha}}\in\mathbb{R}^{|\tilde{\alpha}|}, x_{\tilde{\beta}}\in\mathbb{R}^{|\tilde{\beta}|}, y_{\tilde{I}^c}\in\mathbb{R}^{|\tilde{I}^c|}} & \tilde{\varphi}(x_{\tilde{\alpha}}, x_{\tilde{\beta}}) + \lambda\tilde{q}(y_{\tilde{I}^c}) \\ \text{s.t.} & (B_{\tilde{I}^c\tilde{\alpha}} + B_{\tilde{I}^c\tilde{\gamma}} M_{\tilde{\gamma}\tilde{\alpha}}) x_{\tilde{\alpha}} + B_{\tilde{I}^c\tilde{\beta}} x_{\tilde{\beta}} - y_{\tilde{I}^c} = 0, \end{array}$$

where

$$\tilde{\varphi}(x_{\tilde{\alpha}}, x_{\tilde{\beta}}) = f(\dot{x}), \quad \tilde{q}(y_{\tilde{I}^c}) = p(\dot{y}).$$

Here

$$\dot{x}_{\tilde{\alpha}} = x_{\tilde{\alpha}}, \quad \dot{x}_{\tilde{\beta}} = x_{\tilde{\beta}}, \quad \dot{x}_{\tilde{\gamma}} = M_{\tilde{\gamma}\tilde{\alpha}} x_{\tilde{\alpha}},$$

and

$$\dot{y}_i = \begin{cases} y_i & \text{if } i \in \tilde{I}^c, \\ 0 & \text{if } i \in \tilde{I}. \end{cases}$$

Then, there exists a $\tilde{\theta} \in \mathbb{R}^{|\tilde{I}^c|}$ such that the following KKT conditions are satisfied:

$$\begin{cases} (\nabla f(\tilde{x}))_{\tilde{\alpha}} + M_{\tilde{\gamma}\tilde{\alpha}}^T (\nabla f(\tilde{x}))_{\tilde{\gamma}} + (B_{\tilde{I}^c\tilde{\alpha}} + B_{\tilde{I}^c\tilde{\gamma}} M_{\tilde{\gamma}\tilde{\alpha}})^T \tilde{\theta} = 0, \\ (\nabla f(\tilde{x}))_{\tilde{\beta}} + B_{\tilde{I}^c\tilde{\beta}}^T \tilde{\theta} = 0, \quad \tilde{\theta} \in \lambda(\partial p(\tilde{y}))_{\tilde{I}^c}, \\ (B_{\tilde{I}^c\tilde{\alpha}} + B_{\tilde{I}^c\tilde{\gamma}} M_{\tilde{\gamma}\tilde{\alpha}})\tilde{x}_{\tilde{\alpha}} + B_{\tilde{I}^c\tilde{\beta}}\tilde{x}_{\tilde{\beta}} - \tilde{y}_{\tilde{I}^c} = 0. \end{cases} \tag{2.19}$$

Again, the key is to construct a dual pair $(\tilde{v}, \tilde{s})$ from the KKT system (2.19) such that $(\tilde{x}, \tilde{y}, \tilde{v}, \tilde{s})$ is a solution to (2.18). Fortunately, by Assumption 2.2 and the fact $\tilde{I}^c = \{i \in [m] \mid \tilde{y}_i \neq 0\}$, we have

$$\tilde{v}_{\tilde{I}^c} = (\partial(\lambda p(\tilde{y})))_{\tilde{I}^c} = \tilde{\theta}. \tag{2.20}$$

Thus, by the uniqueness of $\tilde{v}_{\tilde{I}^c}$, the second equation of (2.18) must be satisfied.

Similarly, we construct $(\tilde{v}_{\tilde{I}}, \tilde{s})$ as follows:

$$\tilde{v}_{\tilde{I}} = (\tilde{v}_{\tilde{I}})_0 + \tilde{d}, \quad \tilde{s} = \tilde{v}_{\tilde{I}} - \Pi_{(\partial(\lambda p(\bar{y})))_{\tilde{I}}}(\tilde{v}_{\tilde{I}}), \tag{2.21}$$

where

$$(\tilde{v}_{\tilde{I}})_0 = -B_{\tilde{I}\tilde{\gamma}}(B_{\tilde{I}\tilde{\gamma}}^T B_{\tilde{I}\tilde{\gamma}})^{-1}\big((\nabla f(\tilde{x}))_{\tilde{\gamma}} + B_{\tilde{I}^c\tilde{\gamma}}^T \tilde{v}_{\tilde{I}^c}\big)$$

and $\tilde{d}$ is an optimal solution to the following auxiliary optimization problem:

$$\begin{array}{ll} \min_{d \in \mathbb{R}^{|\tilde{I}|}} & \frac{1}{2}\|((\tilde{v}_{\tilde{I}})_0 + d) - \Pi_{(\partial(\lambda p(\tilde{y})))_{\tilde{I}}}((\tilde{v}_{\tilde{I}})_0 + d)\|^2 \\ \text{s.t.} & d \in \mathrm{Null}(B_{\tilde{I}\tilde{\gamma}}^T). \end{array} \tag{2.22}$$

For the above constructed $(\tilde{v}, \tilde{s})$, it has a nice property to be summarized in the following theorem. It shows that the constructed dual variable $\tilde{v}$ can certify the ϵ-optimality of $\tilde{x}$.

THEOREM 2.8. *For a given $\epsilon > 0$, if the current solution $\tilde{x}$ obtained by solving $(RP_\lambda(I))$ is an optimal solution to the perturbed optimization problem*

$$\min_{x \in \mathbb{R}^n} \quad f(x) + \lambda p(Bx) + \langle x, \tilde{\delta} \rangle, \tag{2.23}$$

where $\tilde{\delta} \in \mathbb{R}^n$ is a latent error vector such that $\|\tilde{\delta}\| \leq \frac{\epsilon}{1+2L_{\tilde{\gamma}}}$, and the constant $L_{\tilde{\gamma}} = \|B_{\tilde{I}\tilde{\gamma}}(B_{\tilde{I}\tilde{\gamma}}^T B_{\tilde{I}\tilde{\gamma}})^{-1}\|$, then, we must have

$$\|R_\lambda(\tilde{x}, \tilde{y}, \tilde{v})\| \leq \epsilon,$$

where $\tilde{y} = B\tilde{x}$ and $\tilde{v}$ is constructed in (2.20), (2.21), *and* (2.22). *Thus we certify the ϵ-optimality of $\tilde{x}$.*

Proof. If $\tilde{x}$ is an optimal solution to (2.23), then $(\tilde{x}, \tilde{y})$ is an optimal solution to

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} & f(x) + \lambda p(y) + \langle x, \tilde{\delta} \rangle \\ \text{s.t.} & Bx - y = 0. \end{array} \tag{2.24}$$

Then, there exists a $\tilde{z} \in \mathbb{R}^m$, which satisfies the following KKT conditions:

$$\begin{cases} (\nabla f(\tilde{x}))_{\tilde{\alpha}} + B^T_{\tilde{I}\tilde{\alpha}}\tilde{z}_{\tilde{I}} + B^T_{\tilde{I}^c\tilde{\alpha}}\tilde{z}_{\tilde{I}^c} + \tilde{\delta}_{\tilde{\alpha}} = 0, \\ (\nabla f(\tilde{x}))_{\tilde{\beta}} + B^T_{\tilde{I}^c\tilde{\beta}}\tilde{z}_{\tilde{I}^c} + \tilde{\delta}_{\tilde{\beta}} = 0, \\ (\nabla f(\tilde{x}))_{\tilde{\gamma}} + B^T_{\tilde{I}\tilde{\gamma}}\tilde{z}_{\tilde{I}} + B^T_{\tilde{I}^c\tilde{\gamma}}\tilde{z}_{\tilde{I}^c} + \tilde{\delta}_{\tilde{\gamma}} = 0, \\ \tilde{z} \in \partial(\lambda p(\tilde{y})), \\ B\tilde{x} - \tilde{y} = 0. \end{cases} \tag{2.25}$$

By Assumption 2.2 and the fact $\tilde{I}^c = \{i \in [m] \mid \tilde{y}_i \neq 0\}$, $(\partial(\lambda p)(\tilde{y}))_{\tilde{I}^c}$ is a singleton. Thus we must have

$$\tilde{z}_{\tilde{I}^c} = \tilde{v}_{\tilde{I}^c}.$$

Therefore, $\bar{v}_{\tilde{I}} = \tilde{z}_{\tilde{I}} - B_{\tilde{I}\tilde{\gamma}}(B^T_{\tilde{I}\tilde{\gamma}}B_{\tilde{I}\tilde{\gamma}})^{-1}\tilde{\delta}_{\tilde{\gamma}}$ is a solution to

$$(\nabla f(\tilde{x}))_{\tilde{\gamma}} + B^T_{\tilde{I}\tilde{\gamma}}v_{\tilde{I}} + B^T_{\tilde{I}^c\tilde{\gamma}}\tilde{v}_{\tilde{I}^c} = 0.$$

Thus, there exists a vector $\bar{d} \in \mathrm{Null}(B^T_{\tilde{I}\tilde{\gamma}})$ such that

$$\bar{v}_{\tilde{I}} = (\tilde{v}_{\tilde{I}})_0 + \bar{d}.$$

Since $\tilde{d}$ is a solution to (2.22) and $\tilde{v}_{\tilde{I}} = (\tilde{v}_{\tilde{I}})_0 + \tilde{d}$, we have

$$\begin{aligned} \|\tilde{s}\| &= \|\tilde{v}_{\tilde{I}} - \Pi_{(\partial(\lambda p(\tilde{y})))_{\tilde{I}}}(\tilde{v}_{\tilde{I}})\| \\ &\leq \|\bar{v}_{\tilde{I}} - \Pi_{(\partial(\lambda p(\tilde{y})))_{\tilde{I}}}(\bar{v}_{\tilde{I}})\| \\ &= \|(\tilde{z}_{\tilde{I}} - B_{\tilde{I}\tilde{\gamma}}(B^T_{\tilde{I}\tilde{\gamma}}B_{\tilde{I}\tilde{\gamma}})^{-1}\tilde{\delta}_{\tilde{\gamma}}) - \Pi_{(\partial(\lambda p(\tilde{y})))_{\tilde{I}}}(\tilde{z}_{\tilde{I}} - B_{\tilde{I}\tilde{\gamma}}(B^T_{\tilde{I}\tilde{\gamma}}B_{\tilde{I}\tilde{\gamma}})^{-1}\tilde{\delta}_{\tilde{\gamma}})\| \\ &= \| - B_{\tilde{I}\tilde{\gamma}}(B^T_{\tilde{I}\tilde{\gamma}}B_{\tilde{I}\tilde{\gamma}})^{-1}\tilde{\delta}_{\tilde{\gamma}} + (\Pi_{(\partial(\lambda p(\tilde{y})))_{\tilde{I}}}(\tilde{z}_{\tilde{I}}) \\ &\quad -\Pi_{(\partial(\lambda p(\tilde{y})))_{\tilde{I}}}(\tilde{z}_{\tilde{I}} - B_{\tilde{I}\tilde{\gamma}}(B^T_{\tilde{I}\tilde{\gamma}}B_{\tilde{I}\tilde{\gamma}})^{-1}\tilde{\delta}_{\tilde{\gamma}}))\| \\ &\leq 2\|B_{\tilde{I}\tilde{\gamma}}(B^T_{\tilde{I}\tilde{\gamma}}B_{\tilde{I}\tilde{\gamma}})^{-1}\tilde{\delta}_{\tilde{\gamma}}\| \\ &\leq 2L_{\tilde{\gamma}}\|\tilde{\delta}_{\tilde{\gamma}}\|. \end{aligned}$$

On the other hand, by the construction, we know that $(\tilde{x}, \tilde{y}, \tilde{v}, \tilde{s})$ satisfies the following KKT system:

$$\begin{cases} (\nabla f(\tilde{x}))_{\tilde{\alpha}} + B^T_{\tilde{I}\tilde{\alpha}}\tilde{v}_{\tilde{I}} + B^T_{\tilde{I}^c\tilde{\alpha}}\tilde{v}_{\tilde{I}^c} + \tilde{\delta}_{\tilde{\alpha}} = 0, \\ (\nabla f(\tilde{x}))_{\tilde{\beta}} + B^T_{\tilde{I}^c\tilde{\beta}}\tilde{v}_{\tilde{I}^c} + \tilde{\delta}_{\tilde{\beta}} = 0, \\ (\nabla f(\tilde{x}))_{\tilde{\gamma}} + B^T_{\tilde{I}\tilde{\gamma}}\tilde{v}_{\tilde{I}} + B^T_{\tilde{I}^c\tilde{\gamma}}\tilde{v}_{\tilde{I}^c} = 0, \\ \tilde{v}_{\tilde{I}} - \tilde{s} \in (\partial(\lambda p(\tilde{y})))_{\tilde{I}}, \quad \tilde{v}_{\tilde{I}^c} \in (\partial(\lambda p(\tilde{y})))_{\tilde{I}^c}. \\ B\tilde{x} - \tilde{y} = 0. \end{cases}$$

Then, we have

$$\begin{aligned} \|R_\lambda(\tilde{x}, \tilde{y}, \tilde{v})\| &= \|(\nabla f(\tilde{x}) + B^T\tilde{v}, \quad \tilde{y} - \mathrm{Prox}_{\lambda p}(\tilde{y} + \tilde{v}), \quad B\tilde{x} - \tilde{y})\| \\ &\leq \|(\tilde{\delta}_{\tilde{\alpha}}, \tilde{\delta}_{\tilde{\beta}}, 0)\| + \|\tilde{s}\| \\ &\leq \|\tilde{\delta}\| + 2L_{\tilde{\gamma}}\|\tilde{\delta}\| \\ &\leq \epsilon. \end{aligned}$$

This completes the proof of the theorem. ∎

Now, we present the EAS technique in Algorithm 2.2. As a byproduct of Theorem 2.4, Theorem 2.6, and Theorem 2.8, we have the following theorem.

Algorithm 2.2 An EAS for solving (P_λ) with a fixed $\lambda > 0$

Input: a given hyperparameter $\lambda > 0$ and a given tolerance $\epsilon > 0$.
Output: $(x^*(\lambda), y^*(\lambda), z^*(\lambda))$.
Initialization: Generate an initial index set by a predefined initialization strategy: $I^0(\lambda) \subseteq [m]$.
for $i = 0, 1, 2, \ldots$ **do**
  1. For the given index set $I^i(\lambda)$, construct the index partition $\{\alpha^i, \beta^i, \gamma^i\}$ and the corresponding $M_{\gamma^i\alpha^i}$.
  2. Apply any well designed algorithm to solving problem $(RP_\lambda(I))$ with the constructed $\{I^i(\lambda), \alpha^i, \beta^i, \gamma^i, M_{\gamma^i\alpha^i}\}$, and obtain an inexact solution $(\hat{x}^i_{\alpha_i}, \hat{x}^i_{\beta_i}, \hat{y}^i_{(I^i)^c(\lambda)}, \hat{\xi}^i)$ which satisfies the corresponding KKT system (2.14) with the latent error terms $(\hat{\delta}^i_1, \hat{\delta}^i_2, \hat{\delta}^i_3)$ such that $\|\hat{\delta}^i_1\| + \|\hat{\delta}^i_2\| + \|\hat{\delta}^i_3\| \le \epsilon$.
  3. Recover a solution $(\bar{x}^i, \bar{y}^i)$ by (2.5).
  if $i > 1$ and $|F_\lambda(\bar{x}^i) - F_\lambda(\bar{x}^{i-1})| \le \epsilon$ **then**
    Define $(\tilde{x}^i, \tilde{y}^i) = (\bar{x}^i, B\bar{x}^i)$ and $\tilde{I}^i = \{i \in [m] \mid \tilde{y}_i = 0\}$. Construct $\{\tilde{\alpha}^i, \tilde{\beta}^i, \tilde{\gamma}^i, M_{\tilde{\gamma}^i\tilde{\alpha}^i}\}$.
    Construct $(\tilde{v}^i, \tilde{s}^i)$ by (2.20), (2.21), and (2.22).
    if $\|R_\lambda(\tilde{x}^i, \tilde{y}^i, \tilde{v})\| \le \epsilon$ **then**
      Set $(x^*(\lambda), y^*(\lambda), z^*(\lambda)) = (\tilde{x}^i, \tilde{y}^i, \tilde{v}^i)$.
      break.
    end if
  end if
  4. Recover a pair $(\bar{u}^i, \bar{w}^i)$ by (2.7) and (2.10), respectively.
  if $\|R_\lambda(\bar{x}^i, \bar{y}^i, \bar{u}^i)\| \le \epsilon$ **then**
    Set $(x^*(\lambda), y^*(\lambda), z^*(\lambda)) = (\bar{x}^i, \bar{y}^i, \bar{u}^i)$.
    break.
  else
    Create $J^i(\lambda)$:

$$J^i(\lambda) = \{j \in I^i(\lambda) \mid \bar{u}^i_j \notin \partial(\lambda p(\bar{y}^i))_j\}. \tag{2.26}$$

    Update $I^{i+1}(\lambda)$ as

$$I^{i+1}(\lambda) \leftarrow I^i(\lambda) \backslash J^i(\lambda).$$

  end if
end for
return $(x^*(\lambda), y^*(\lambda), z^*(\lambda))$.

THEOREM 2.9. *For a given $\epsilon > 0$, Algorithm 2.2 is guaranteed to converge in a finite number of iterations. The number of sieving iterations of Algorithm 2.2 is no more than that of Algorithm 2.1. Moreover, the obtained pair $(x^*(\lambda), y^*(\lambda), z^*(\lambda))$ is a solution to (P_λ) in the sense that*

$$\|R_\lambda(x^*(\lambda), y^*(\lambda), z^*(\lambda))\| \le \epsilon.$$

Remark 2.10. We close this subsection by making some remarks.

1. The EAS technique described in Algorithm 2.2 is a rigorous implementation of the aforementioned principal idea which simultaneously answers all the five questions posed earlier in section 2.
2. A natural question is, Why should we still perform the sieving based on $\bar{u}$ instead of $\tilde{v}$ directly? The reason is that if we define

$$\tilde{J}(\lambda) = \{j \in \tilde{I} \mid \tilde{v}_j \notin (\partial(\lambda p(\tilde{y})))_j\},$$

assuming that $\tilde{J}(\lambda) \neq \emptyset$, we cannot guarantee that $\tilde{J}(\lambda) \bigcap I \neq \emptyset$, which is required to update the index set I.

3. The EAS algorithm has an additional procedure to certify the ϵ-optimality of the obtained solution. Thus, we may stop the sieving procedure earlier, compared to Algorithm 2.1. It is a natural idea that we only try to certify the ϵ-optimality of the current obtained solution when $|F_\lambda(\bar{x}^i) - F_\lambda(\bar{x}^{i-1})| < \epsilon$. The reason we use the difference of consecutive function values instead of the solution vectors is because the optimal solutions of (P_λ) may not be unique, but they all have the same objective function value.
4. In practice, the AS technique is sometimes better than the EAS technique in terms of running time. But the EAS technique is the one with a better theoretical guarantee. Detailed empirical comparison of these two techniques can be found in section 4.

2.4. An accelerated proximal gradient algorithm for dual variable recovery. As aforementioned, a key step to apply the AS (or EAS) technique is recovering the dual variable $\bar{u}$ (or $\tilde{v}$) via solving the optimization problem (2.11) (or (2.22)). In this paper, we adopt the accelerated proximal gradient (APG) algorithm [2, 22] to solve the optimization problems (2.11) and (2.22). Since the optimization problem (2.22) has the same form as (2.11), we use the problem (2.11) as an example.

First of all, we could rewrite the constrained optimization problem (2.11) equivalently as

$$\min_d h(d) + \delta_{\mathrm{Null}(B_{I_\gamma}^T)}(d), \tag{2.27}$$

where $h(d) = \frac{1}{2}\|((\bar{u}_I)_0 + d) - \Pi_{\partial(\lambda p(\bar{y}))_I}((\bar{u}_I)_0 + d)\|^2$ and $\delta_{\mathrm{Null}(B_{I_\gamma}^T)}(\cdot)$ is the indicator function of the null space of $B_{I_\gamma}^T$.

In order to apply the APG algorithm, we need to derive the proximal mapping of the indicator function $\delta_{\mathrm{Null}(B_{I_\gamma}^T)}(\cdot)$, which is the projection operator onto the null space of $B_{I_\gamma}^T$. Since $B_{I_\gamma}^T$ is of full row rank, the projection of a given vector $a \in \mathbb{R}^{|I|}$ onto the null space of $B_{I_\gamma}^T$ is computed by

$$\Pi_{\mathrm{Null}(B_{I_\gamma}^T)}(a) = (I - B_{I\gamma}(B_{I_\gamma}^T B_{I\gamma})^{-1} B_{I_\gamma}^T)a.$$

On the other hand, the function $h(\cdot)$ is continuously differentiable, and the gradient of $h(\cdot)$ is

$$\nabla h(d) = ((\bar{u}_I)_0 + d) - \Pi_{\partial(\lambda p(\bar{y}))_I}((\bar{u}_I)_0 + d) = \Pi_{(\partial(\lambda p(\bar{y}))_I)^\circ}((\bar{u}_I)_0 + d).$$

Here, $(\partial(\lambda p(\bar{y}))_I)^\circ$ is the polar of the closed convex set $\partial(\lambda p(\bar{y}))_I$, and the second equality comes from the Moreau identity [20]. Thus, $\nabla h(\cdot)$ is Lipschitz continuous with modulus 1 [40]. The APG algorithm for solving the optimization problem (2.27) is shown in Algorithm 2.3.

It is well known that the sequence $\{d^k\}$ generated by the APG algorithm has the following $O(1/k^2)$ complexity [2, 22].

THEOREM 2.11. *Let $\{d^k\}$ and $\{t_k\}$ be the sequences generated by Algorithm* 2.3. *Then for any $k \geq 1$, we have*

$$h(d^k) - h(d^*) \leq \frac{2\|d^*\|^2}{(k+1)^2}, \tag{2.28}$$

where d^* is any optimal solution to (2.27).

Algorithm 2.3 APG algorithm for (2.27)

Input: $\epsilon > 0$ and maxiter.
Output: $\bar{d}$.
Initialization: $L = 1$, $d^0 = 0$, $\hat{d}^1 = d^0$, $k = 0$, and $t_1 = 1$.
while $k <$ maxiter **do**
  $k = k + 1$,
  $d^k = \Pi_{\mathrm{Null}(B_{I_\gamma}^T)}(\hat{d}^k - \frac{1}{L}\nabla h(\hat{d}^k))$.
  if $\max(\|d^k - d^{k-1}\|, \|((\bar{u}_I)_0 + d^k) - \Pi_{\partial(\lambda p(\bar{y}))_I}((\bar{u}_I)_0 + d^k)\|) \le \epsilon$ **then**
    break.
  end if
  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,
  $\hat{d}^{k+1} = d^k + (\frac{t_k - 1}{t_{k+1}})(d^k - d^{k-1})$.
end while
$\bar{d} = d^k$.
return $\bar{d}$.

Remark 2.12. We make some remarks to close this subsection.

1. The optimal solution set for (2.27) is nonempty under the mild assumption that $(\partial(\lambda p(\bar{y})))_I$ is nonempty and compact. This assumption is satisfied for any $\bar{y} \in \mathbb{R}^m$ and nonempty set $I \subseteq [m]$ and for many regularization functions, such as ℓ_1 norm, ℓ_2 norm, ℓ_∞ norm, and so on.
2. The computational cost for solving (2.27) in the AS and EAS techniques is affordable. We explain some insights behind this remark. On the one hand, in the EAS technique, if we do obtain an optimal solution of (P_λ) via solving the current subproblem $(P_\lambda(I))$, then we must have $h(d^*) < \frac{\epsilon^2}{2}$ by Theorem 2.8. Moreover, $\|d^*\|$ must be relatively small. By the above complexity result, we could obtain an inexact solution to the problem (2.27) in several cheap iterations. On the other hand, if the objective function value of (2.27) is still large after several iterations (say 10 iterations), we can terminate the algorithm since this phenomenon indicates that we have not yet obtained an optimal solution to the problem (P_λ). In other words, the current index set I is incorrect, and we need to update it by removing violating indices. In short, although we need to solve an additional optimization problem, we only need to run APG for several iterations.
3. The main computational cost for each iteration of APG is from two projections. On the one hand, for most of the commonly used regularization functions p (for example, ℓ_1 norm, ℓ_2 norm), the projection of a given vector onto the subdifferential set is cheap. On the other hand, in order to compute the projection onto the null space of $B_{I_\gamma}^T$, the main computational cost is from computing $(B_{I_\gamma}^T B_{I_\gamma})^{-1}$. However, as we mentioned earlier, the matrix B is very sparse in many applications, and the sparse Cholesky decomposition is not costly. Thus, the computational cost for one iteration of APG is affordable, even for large-scale problems. This is also one of the main reasons for us to adopt APG to solve the optimization problem (2.27).

2.5. An AS technique for solution path. Now, we generalize Algorithm 2.1 and Algorithm 2.2 to obtain a solution path for problem (P_λ) with a sequence of parameters $\lambda_1 > \lambda_2 > \cdots > \lambda_l > 0$. If we obtain a solution $(x^*(\lambda_i), y^*(\lambda_i), z^*(\lambda_i))$ for (P_λ) with $\lambda = \lambda_i$, then we can initialize the index set $I^0(\lambda_{i+1})$ in Algorithm 2.1

or Algorithm 2.2 for $\lambda = \lambda_{i+1}$ as

$$I^0(\lambda_{i+1}) := \{k \in [m] \mid |(Bx^*(\lambda_i))_k| < \hat{\epsilon}\}, \tag{2.29}$$

where $\hat{\epsilon} > 0$ is a given tolerance. The algorithm for applying the AS technique (or the EAS technique) to generate a solution path is shown in Algorithm 2.4.

Algorithm 2.4 Generate solution path for (P_λ) with the AS technique (or the EAS technique)

Input: $\epsilon > 0$, $\hat{\epsilon} > 0$ and a sequence $\lambda_1 > \lambda_2 > \cdots > \lambda_l > 0$.
Output: A solution path for (P_λ): $\{(x^*(\lambda_1), y^*(\lambda_1), z^*(\lambda_1)), \ldots, (x^*(\lambda_l), y^*(\lambda_l), z^*(\lambda_l))\}$.

Initialization: Initialize index set $I^0(\lambda_1) \subseteq [m]$ by a predefined initialization strategy.
for $k = 1, 2, \ldots, l$ **do**
  Step 1. Obtain $(x^*(\lambda_k), y^*(\lambda_k), z^*(\lambda_k))$ by calling Algorithm 2.1 (or Algorithm 2.2) with $\{\lambda, \epsilon, I^0(\lambda)\} = \{\lambda_k, \epsilon, I^0(\lambda_k)\}$.
  if $k < l$ **then**
    Step 2. Define
$$I^0(\lambda_{k+1}) := \{j \in [m] \mid |(Bx^*(\lambda_k))_j| < \hat{\epsilon}\}.$$
  end if
end for
return $\{(x^*(\lambda_1), y^*(\lambda_1), z^*(\lambda_1)), \ldots, (x^*(\lambda_l), y^*(\lambda_l), z^*(\lambda_l))\}$.

Remark 2.13. We give some remarks to close this subsection.

1. First, we make some remarks on the choice of $I^0(\lambda_i)$, which is defined by (2.29). Note that for $\lambda = \lambda_i$, Algorithm 2.1 and Algorithm 2.2 can both obtain an ϵ-optimal solution of (P_λ) with any initial index set $I^0(\lambda_i)$. We define $I^0(\lambda_i)$ $(i \geq 2)$ as in (2.29) because the solution for (P_λ) with $\lambda = \lambda_{i-1}$ will provide better prior information. The index set $I^0(\lambda_1)$ is indeed problem dependent; however, since λ_1 is the largest value of λ on the solution path, it will generate a highly structured sparse solution. Thus, we can choose $I^0(\lambda_1)$ in an aggressive way.
2. For the convex clustering model with uniform weights ($w_{ij} = 1,\ 1 \leq i < j \leq N$), it has the agglomeration property [6]. For the convex clustering model with general weights, the agglomeration property may not hold. However, Chi and Steinerberger [5] proved the tree structure of the solution path under affinity weights. In these cases, we know that $\tilde{I}(\lambda_{i+1}) \subseteq \tilde{I}(\lambda_i),\quad i = 1, 2, \ldots, l-1$, where $\tilde{I}(\lambda_i) := \{j \in [m] \mid (y^*(\lambda_i))_j = 0\}$. Thus, the solution $(x^*(\lambda_i), y^*(\lambda_i), z^*(\lambda_i))$ includes prior information for the next problem in the solution path.
3. Another natural question is, Why not solve a solution path in the reverse order (i.e., starting from $\lambda = \lambda_l$)? On the one hand, the agglomeration property may not hold. On the other hand, the solution to (P_λ) with $\lambda = \lambda_l$ is the densest. If we start from this problem, then it could be challenging to choose an appropriate initial index set $I^0(\lambda_l)$.

3. AS and EAS techniques for convex clustering model. In this section, we will show how to apply the AS and EAS techniques to the convex clustering model (1.1).

Denote $\mathcal{E} := \{(i, j) \mid w_{ij} > 0, 1 \leq i < j \leq n\}$. Then $\mathcal{G} = ([n], \mathcal{E})$ forms an undirected graph, and the weighted convex clustering model (1.1) is equivalent to

$$\min_{X\in\mathbb{R}^{d\times N}} \frac{1}{2}\sum_{i=1}^{N}\|X_{:i}-A_{:i}\|_2^2+\lambda\sum_{(i,j)\in\mathcal{E}} w_{ij}\|X_{:i}-X_{:j}\|_p. \tag{3.1}$$

We enumerate the index pairs in $\mathcal{E}$ by the lexicographic order and denote by $l(i,j)$ the pair (i,j). Define the linear map $\mathcal{B}:\mathbb{R}^{d\times N}\to\mathbb{R}^{d\times|\mathcal{E}|}$ as

$$(\mathcal{B}(X))_{:,l(i,j)} = X_{:i}-X_{:j},$$

and define the node-arc incidence matrix $J\in\mathbb{R}^{N\times|\mathcal{E}|}$ as

$$J_{k,l(i,j)} = \begin{cases} 1 & \text{if } k=i,\\ -1 & \text{if } k=j,\\ 0 & \text{otherwise.}\end{cases} \tag{3.2}$$

Then, for any given $X\in\mathbb{R}^{d\times N}$ and $Z\in\mathbb{R}^{d\times|\mathcal{E}|}$, we have

$$\mathcal{B}(X)=XJ,\quad \mathcal{B}^*(Z)=ZJ^T. \tag{3.3}$$

It is clear that the convex clustering model (3.1) is a special case of (1.2).

3.1. A construction of the reduced problem. The main step for constructing the reduced subproblem is to construct the index sets α,β,γ and the corresponding matrix $M_{\gamma\alpha}$. For a given index set

$$I := \{l(i,j)\}\subseteq\{1,2,\ldots,|\mathcal{E}|\}, \tag{3.4}$$

we can construct a subgraph $\hat{\mathcal{G}}\subseteq\mathcal{G}$ with edges $\hat{\mathcal{E}} := \{(i,j)\mid l(i,j)\in I\}$ and all the corresponding nodes. Then, we can decompose the graph $\hat{\mathcal{G}}$ as

$$\hat{\mathcal{G}} = \hat{\mathcal{G}}_1\cup\hat{\mathcal{G}}_2\cup\cdots\cup\hat{\mathcal{G}}_s,$$

where $\hat{\mathcal{G}}_i$ are disjoint connected subgraphs of $\hat{\mathcal{G}}$. Denote the node index set of $\hat{\mathcal{G}}_i$ as $\hat{\mathcal{N}}_i$, and we define

$$\alpha_i = \min\{k\mid k\in\hat{\mathcal{N}}_i\},\quad i=1,2,\ldots,s.$$

Then, we can uniquely determine the index sets α, β, and γ as

$$\alpha=\{\alpha_1,\ldots,\alpha_s\},\quad \beta=[N]\backslash(\hat{\mathcal{N}}_1\cup\cdots\cup\hat{\mathcal{N}}_s),\quad\text{and}\quad \gamma=(\hat{\mathcal{N}}_1\cup\cdots\cup\hat{\mathcal{N}}_s)\backslash\alpha.$$

The index sets α, β, and γ have clear meanings in the convex clustering model. For a given index set $I\subseteq[|\mathcal{E}|]$ and the generated graph $\hat{\mathcal{G}}$, α_i is the index of the selected representative point for the ith cluster identified by the connected component $\hat{\mathcal{G}}_i$. On the other hand, β is the collection of the indices of the isolated clusters which contain only a singleton.

Furthermore, we could have an explicit formula for $M_{\gamma\alpha}\in\mathbb{R}^{|\alpha|\times|\gamma|}$, which is given by

$$(M_{\gamma\alpha})_{ij} = \begin{cases} 1 & \text{if } j\in\hat{\mathcal{N}}_i,\\ 0 & \text{otherwise.}\end{cases}$$

Then

$$X_{:\gamma} = X_{:\alpha}M_{\gamma\alpha},$$

which actually maps the data points indexed by γ to the corresponding representative points with indices in the set α.

4. Numerical experiments. In this section, we demonstrate the efficiency of the proposed AS and EAS techniques via the convex clustering model (3.1) (with $p = 2$). In this paper, we mainly focus on the numerical efficiency of the AS and EAS techniques; readers can refer to [30, 10, 17] and the references therein for the performance of clustering by the convex clustering model (3.1). We test the AS and EAS techniques with AMA [4], ADMM [4], and SSNAL [39], which are three of the most popular algorithms for solving (3.1). Due to the limited length of the paper, we omit the details of these three algorithms but refer the readers to the aforementioned references. In our experiments, by default, we will generate the clustering path with $\lambda =: [10 : -0.2 : 1]$. The weights w_{ij} will be defined by the following Gaussian kernel with k-nearest neighbors (we choose $k = 10$ in our experiments):

$$w_{ij} = \begin{cases} \exp(-\frac{1}{2}\|A_{:i} - A_{:j}\|^2) & \text{if } (i,j) \in \mathcal{E}, \\ 0 & \text{if } (i,j) \notin \mathcal{E}, \end{cases} \tag{4.1}$$

where $\mathcal{E} = \{(i,j) \mid A_{:i}$ is among $A_{:j}$'s k nearest neighbors$\}$.

The choice of $I^0(\lambda_1)$ also has impact on the numerical performance of the AS and EAS techniques. For the convex clustering model, it is more likely that $X^*_{:i}(\lambda_1) = X^*_{:j}(\lambda_1)$ if $\|A_{:i} - A_{:j}\|$ is smaller. Thus, we choose $I^0(\lambda_1)$ as

$$I^0(\lambda_1) = \{l(i,j) \mid (i,j) \in \mathcal{E}, w_{ij} \text{ is among the top } 20\% \text{ largest weights}\}.$$

For a fair comparison with the fast AMA algorithm, in this paper, we terminate all the algorithms based on the relative duality gap:

$$\eta = \tfrac{F_\lambda(X) - D_\lambda(Z)}{1 + |F_\lambda(X)| + |D_\lambda(Z)|} \le \epsilon. \tag{4.2}$$

Here, $\epsilon > 0$ is a given tolerance, and $F_\lambda(X)$ and $D_\lambda(Z)$ are the objective function values of the primal problem (1.2) and the dual problem (D_λ), respectively. We set $\epsilon = 10^{-6}$ in (4.2) and $\hat{\epsilon} =$ 2e-16 in (2.29) by default. All our numerical results are obtained by running MATLAB on a Windows workstation (Intel Xeon E5-2680 @ 2.50GHz).

4.1. Simulated datasets. In this subsection, we provide some numerical results on the simulated two half-moon data, which is one of the most popular datasets for clustering.

First, we revisit the performance of fast AMA [4], ADMM [4], and SSNAL [39] for generating the clustering path directly. We implemented the three algorithms in MATLAB and tried our best to optimize the computations for a fair comparison.[3] We summarize some numerical results in Table 1. We can see that SSNAL is the most efficient algorithm on this dataset. Chi and Lange claimed in [4] that fast AMA is much better than ADMM for solving the convex clustering model. However, we observe some discrepancies in *our own* numerical experiments. Fast AMA could not achieve the accuracy ($\epsilon = 10^{-6}$) when n is relatively large. For a fairer comparison, we revisit the numerical performance of the three algorithms under a low accuracy setting with $\epsilon = 10^{-4}$. Our numerical results show that ADMM is still better than fast AMA. Since fast AMA has difficulty in solving medium-scale convex clustering model, we focus on applying the AS and EAS techniques with ADMM and SSNAL.

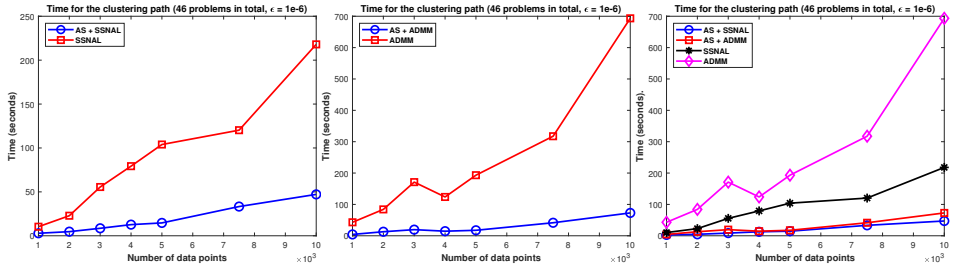
Now, we move on to present the numerical performance of the proposed AS technique. The details can be found in Figure 1. Our numerical results on the two half-moon dataset show that the AS technique could accelerate the SSNAL and the ADMM

[3] Readers can find the code at https://blog.nus.edu.sg/mattohkc/softwares/convexclustering/.

TABLE 1

Numerical results for SSNAL, *ADMM, and fast AMA on the two half-moon dataset with* $k = 10$ *and* $\lambda = [10 : -0.2 : 1]$ *(46 problems in total).*

		Time (s)	# unsolved problems
(d, N)	ϵ	SSNAL | ADMM | Fast AMA	SSNAL | ADMM | Fast AMA
$(2, 1000)$	1e-4	2.2 | 2.4 | 65.3	0 | 0 | 0
	1e-6	10.3 | 43.0 | 218.4	0 | 0 | 2
$(2, 3000)$	1e-4	8.4 | 9.1 | 226.3	0 | 0 | 1
	1e-6	55.5 | 171.1 | 945.5	0 | 0 | 11
$(2, 5000)$	1e-4	15.3 | 17.3 | 387.5	0 | 0 | 1
	1e-6	103.9 | 193.2 | 1479.5	0 | 0 | 29
$(2, 10000)$	1e-4	43.3 | 47.3 | 806.2	0 | 0 | 1
	1e-6	217.9 | 693.4 | 2465.7	0 | 0 | 46



(a) (b) (c)

FIG. 1. *Numerical performance on the two half-moon dataset with* $k = 10$.

TABLE 2

Numerical results for AS and EAS on the two half-moon dataset with $k = 10$ *and* $\lambda = [10 : -0.2 : 1]$ *(46 problems in total). In the table, "a": direct, "b": with AS, "c": with EAS.*

		Time (s)	Sieving rounds	Average problem dimension
(d, N)	Algorithm	a | b | c	a | b | c	a | b | c
$(2, 1000)$	SSNAL	10.3 | 1.97 | 1.97	0 | 10 | 10	1000 | 19 | 19
	ADMM	43.0 | 4.5 | 3.1	0 | 22 | 18	1000 | 39 | 19
$(2, 2000)$	SSNAL	22.8 | 4.3 | 4.3	0 | 18 | 18	2000 | 162 | 162
	ADMM	84.1 | 15.6 | 11.6	0 | 33 | 18	2000 | 175 | 162
$(2, 4000)$	SSNAL	79.3 | 12.6 | 12.1	0 | 44 | 38	4000 | 304 | 301
	ADMM	123.8 | 20.1 | 15.1	0 | 45 | 37	4000 | 463 | 429
$(2, 7500)$	SSNAL	120.2 | 32.4 | 32.4	0 | 44 | 44	7500 | 1276 | 1276
	ADMM	317.4 | 48.4 | 40.8	0 | 51 | 45	7500 | 1336 | 1295
$(2, 10000)$	SSNAL	218.0 | 44.7 | 45.9	0 | 52 | 52	10000 | 1525 | 1525
	ADMM	693.4 | 84.6 | 80.9	0 | 54 | 52	10000 | 1868 | 1820

by up to 4.8 times (Figure 1(a)) and 12.8 times (Figure 1(b)), respectively. With the help of the AS technique, AS+ADMM could even be comparable to AS+SSNAL (Figure 1(c)), which demonstrates the power of the AS technique for capturing the intrinsic structured sparsity of the convex clustering model. Since the AS technique can take advantage of the sparse structure to substantially reduce the dimension of the problem, we can apply the sparse Cholesky decomposition to solving the linear system involved in ADMM in a highly efficient way. This also partially demonstrates that ADMM is efficient to solve-small scale convex clustering problems.

Next, we move on to present the empirical comparison between the AS and EAS techniques on the two half-moon dataset. The results can be found in Table 2. We can

observe that the AS and EAS techniques perform well and the total sieving rounds are small, even for large-scale problems. The average size of the reduced problems is much smaller than that of the original problems. These are the main reasons why the AS and EAS techniques can accelerate the algorithms. On the other hand, the EAS technique could potentially stop the sieving procedure early; thus the EAS technique can have fewer sieving rounds than the AS technique. This phenomenon is indeed observed in the numerical experiments. The numerical results are consistent with our expectation.

Remark 4.1. We close this subsection by making some remarks.

1. From the numerical results in Table 2, we can observe two phenomena: (1) The average problem dimensions of AS/EAS + SSNAL are smaller than those of AS/EAS +ADMM. (2) Compared to AS, EAS can further accelerate ADMM, but not so much for SSNAL. Here, we try to give some explanations. On the one hand, although we set the same tolerance to terminate both SSNAL and ADMM, SSNAL will usually achieve higher accuracy with respect to the relative KKT residual due to its faster convergence rate. The more accurate solution will help the sieving procedure to be more effective. On the other hand, SSNAL is efficient for solving the convex clustering model. As a result, the computational cost of a few more iterations of SSNAL could be comparable to the computational cost of the ϵ-optimality certification procedure of the EAS technique.
2. One may naturally expect the AS and EAS techniques to be more powerful when λ is larger. To verify this expectation, we generate a new clustering path with larger values of λ. The results can be found in Table 3.

4.2. Real datasets. In this subsection, we will present the performance of the AS and EAS techniques for generating the clustering path on the Modified National Institute of Standards and Technology (MNIST) database [15]. We adopt the preprocessing method described in [19], which applies a one hidden layer linear neural network to preprocess the raw images. Then, we apply the convex clustering model (3.1) to the preprocessed data. Our experiments are on the testing set of MNIST data, and the dimension of the preprocessed data is 10×10000. The details can be found in Table 4.

TABLE 3

Numerical results for AS and EAS on the two half-moon dataset with $k = 10$ and $\lambda = [20 : -0.2 : 10]$ (51 problems in total). In the table, "a": direct, "b": with AS, "c": with EAS.

		Time (s)	Sieving rounds	Average problem dimension
(d, N)	Algorithm	a | b | c	a | b | c	a | b | c
$(2, 7500)$	SSNAL	47.0 | 19.7 | 19.7	0 | 34 | 34	7500 | 481 | 481
	ADMM	49.5 | 25.3 | 20.6	0 | 47 | 35	7500 | 505 | 482
$(2, 10000)$	SSNAL	107.8 | 29.0 | 29.0	0 | 33 | 33	10000 | 588 | 588
	ADMM	117.5 | 42.1 | 37.9	0 | 34 | 33	10000 | 714 | 693

TABLE 4

Numerical performance on the MNIST dataset with $k = 10$ and $\lambda = [10 : -0.2 : 1]$ (46 problems in total).

	SSNAL	ADMM
	Direct | With AS | With EAS	Direct | With AS | With EAS
Time (seconds)	1207.7 | 156.3 | 157.3	1823.8 | 128.5 | 132.2
Total sieving round	0 | 45 | 45	0 | 47 | 47
Average problem dimension	10000 | 1377 | 1377	10000 | 1389 | 1389

From the results, we observe that the AS technique could accelerate the ADMM by up to 14.2 times and the SSNAL by up to 7.7 times. It is understandable that AS could be more attractive for ADMM, since the second-order sparsity embedded in the algorithm SSNAL has partially captured the structured sparsity already. Moreover, since the EAS technique does not reduce the sieving iterations on this dataset, compared to the AS technique, the EAS technique will spend more time than the AS technique.

5. Conclusion. In this paper, we propose an AS technique and an EAS technique, which can be applied to various optimization algorithms for convex optimization problems with structured sparsity. The proposed techniques can accelerate optimization algorithms by reducing the dimension of the problems that need to be solved. Numerical performance on the convex clustering model has demonstrated the high efficiency of the proposed dimension reduction techniques. We also established a finite convergence property of the AS and EAS techniques. To better demonstrate the generality of the AS and EAS techniques, we conduct additional numerical experiments on the overlapping group lasso regression model [11]; more details can be found in Appendix A. However, we should note that, in the worst case, the AS and EAS techniques may sieve all the indices. But based on our empirical evaluation, one can say that the AS and EAS techniques work well in practice. As a future research topic, we will make efforts to analyze the average-case complexity of the AS and EAS techniques.

Appendix A. Additional numerical results on overlapping group lasso regression problem. We conduct additional numerical experiments on the overlapping group lasso regression model [11] to further demonstrate the applicability and generality of the AS and EAS techniques. The overlapping group lasso regression model has the following form:

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{2}\|Ax-b\|^2+\lambda\sum_{l=1}^{L} w_l\|x_{G_l}\|_2, \tag{A.1}$$

where $A\in\mathbb{R}^{m\times n}$ and $b\in\mathbb{R}^m$ are given data, $w_l>0$ are given weights and $\lambda>0$ is the hyperparameter, and $G_l\subseteq\{1,2,\ldots,n\}$ is the set of indices for the lth group. Here we assume that $G_1\cup G_2\cup\cdots\cup G_L=\{1,2,\ldots,n\}$ and overlapping indices are allowed. In other words, there may exist $1\le i<j\le L$ such that $G_i\bigcap G_j\neq\emptyset$. We denote the cardinality of the set G_l as $|G_l|$. We denote $s=\sum_{i=1}^{L}|G_l|$. We set $w_l=\sqrt{|G_l|}$.

We can rewrite (A.1) in a compact form as

$$\min_{x\in\mathbb{R}^n} \quad F_\lambda(x)=f(Ax)+\lambda p(Bx), \tag{A.2}$$

where $f(Ax)=\frac{1}{2}\|Ax-b\|^2$, $p(Bx)=\sum_{l=1}^{L} w_l\|B_lx\|_2$, $B=[B_1^T,B_2^T,\ldots,B_L^T]^T$, and $B_l\in\mathbb{R}^{|G_l|\times n}$ are defined as

$$(B_l)_{ij}=\begin{cases}1 & \text{if } j=i_j^l,\\ 0 & \text{otherwise.}\end{cases}$$

Here, without loss of generality, we denote $G_l=\{i_1^l,\ldots i_{|G_l|}^l\}$ and $i_1^l\le\cdots\le i_{|G_l|}^l$.

The dual problem of (A.2) is given by

$$\begin{array}{ll}\max_{u\in\mathbb{R}^m,v\in\mathbb{R}^s} & D_\lambda(u,v)=-f^*(u)-\lambda p^*(v/\lambda)\\ \text{s.t.} & A^Tu+B^Tv=0,\end{array} \tag{A.3}$$

where f^* and p^* are the conjugates of f and p, respectively.

A.1. A construction of the reduced problem. For any given index set $I \subseteq [s]$, we can construct the index set α, β and γ as follows:

$$\alpha = \emptyset,\ \beta = \left\{ j \in [n] \mid \sum_{i \in I} B_{ij} = 0 \right\},\ \gamma = [n] \backslash \beta.$$

Moreover, $x_\gamma = 0$, and it is not necessary to construct the matrix $M_{\gamma\alpha}$. Without loss of generality, we can remove the zero rows of $B_{I^c\beta}$ by redefining the index set I as

$$I = \left\{ i \in [s] \mid \sum_{j=1}^{|\gamma|} (B_{:\gamma})_{ij} = 1 \right\}.$$

A.2. Numerical results. We demonstrate the numerical efficiency of the proposed AS and EAS techniques on the Columbia Object Image Library (COIL-100) dataset [21], which contains color images of 100 objects. The dataset contains 72 images taken from different angles for each object. In particular, we select object 10 in our experiments. We denote $D \in \mathbb{R}^{72\times 49152}$ as the matrix representation of the 72 images, where each row of D is for one image. Then, we generate the response vector $b \in \mathbb{R}^{72}$ by randomly choosing a feature (column) in D, and the rest of the features (columns) are concatenated to be the design matrix $A \in \mathbb{R}^{72\times 49151}$. We generate overlapping groups where each group contains 20 features and consecutive groups overlap by 5 features; the last group will contain 10 features due to the number of features for this dataset. We can generate overlapping groups for the image dataset because the pixels of a natural image are approximately blockwise constant. We apply an ADMM algorithm [9] to solving the dual problem (A.3). We terminate all algorithms based on the relative duality gap and dual feasibility:

$$\eta = \max\left\{ \frac{|F_\lambda(x) - D_\lambda(u,v)|}{1 + |F_\lambda(x)| + |D_\lambda(u,v)|},\ \|A^T u + B^T v\| \right\} \le \epsilon.$$

We set $\epsilon = 10^{-5}$ and $\hat{\epsilon} = 10^{-10}$ in Algorithm 2.4. We generate a solution path with $\lambda \in \{0.9^0, 0.9^1, \ldots, 0.9^{30}\}$. Now, we describe the construction of $I^0(\lambda_1)$. We set the number of initial active features (which is the cardinality of the index set β) to be $\lceil 0.01 * n \rceil$. Since this is a linear regression problem, we choose the initial active features based on the correlation test between each feature vector $A_{:i}$ and the response vector b. That is, we compute $c_i^0 := \frac{|\langle A_{:i}, b\rangle|}{\|A_{:i}\|\|b\|}$ for $i = 1, 2, \ldots, n$ and choose the initial index set β and $I^0(\lambda_1)$ as

$$\beta := \{ i \in [n] \mid c_i^0 \text{ is among the first } \lceil 0.01 * n \rceil \text{ largest values in } c_1^0, \ldots, c_n^0 \}$$

and

$$I^0(\lambda_1) := \left\{ i \in [s] \mid \sum_{j=1}^{|\beta|} (B_{:\beta})_{ij} = 0 \right\}.$$

To get more robust results, we run the experiment 5 times (randomly choosing the response vector b each time) and report the performance in Table 5. The numerical results show that the AS and EAS techniques can accelerate ADMM by more than 100 times for generating the solution path of the overlapping group lasso model.

TABLE 5

Numerical results for overlapping group lasso regression model on the COIL-100 dataset with $\lambda \in \{0.9^0, 0.9^1, \ldots, 0.9^{30}\}$ *(31 problems in total). "i_b" is the column index of the response vector b in the matrix D. Input problem dimensions* $(m, n, s) = (72, 49151, 65531)$*. In the table, "a": direct, "b": with AS, "c": with EAS.*

	Time (s)	Sieving rounds	Average problem dimensions (n, s)
i_b	a | b | c	a | b | c	a | b | c
1129	5141.2 | 86.7 | 86.8	0 | 21 | 21	(49151, 65531) | (267, 373) | (267,373)
1793	6763.2 | 67.0 | 68.3	0 | 15 | 15	(49151, 65531) | (235, 323) | (235, 323)
6834	5952.6 | 39.0 | 39.5	0 | 18 | 18	(49151, 65531) | (261, 362) | (261,362)
10493	2755.7 | 109.8 | 110.0	0 | 34 | 34	(49151, 65531) | (302, 413) | (302, 413)
18230	5366.4 | 38.5 | 38.8	0 | 13 | 13	(49151, 65531) | (211, 289) | (211, 289)

Acknowledgments. We are grateful to the associate editor and the two referees for their helpful suggestions and comments on this paper.

REFERENCES

[1] D. ARTHUR AND S. VASSILVITSKII, `k-means++`*: The advantages of careful seeding*, in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2007, pp. 1027–1035.

[2] A. BECK AND M. TEBOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[3] T. CHEN, F. E. CURTIS, AND D. P. ROBINSON, *A reduced-space algorithm for minimizing* ℓ_1*-regularized convex functions*, SIAM J. Optim., 27 (2017), pp. 1583–1610.

[4] E. C. CHI AND K. LANGE, *Splitting methods for convex clustering*, J. Comput. Graph. Statist., 24 (2015), pp. 994–1013.

[5] E. C. CHI AND S. STEINERBERGER, *Recovering trees with convex clustering*, SIAM J. Math. Data Sci., 1 (2019), pp. 383–407.

[6] J. CHIQUET, P. GUTIERREZ, AND G. RIGAILL, *Fast tree inference with weighted fusion penalties*, J. Comput. Graph. Statist., 26 (2017), pp. 205–216.

[7] F. E. CURTIS, Y. DAI, AND D. P. ROBINSON, *A subspace acceleration method for minimization involving a group sparsity-inducing regularizer*, SIAM J. Optim., 32 (2022), pp. 545–572.

[8] J. FAN AND J. LV, *Sure independence screening for ultrahigh dimensional feature space*, J. R. Stat. Soc. Ser. B Stat. Methodol., 70 (2008), pp. 849–911.

[9] M. FAZEL, T. K. PONG, D. F. SUN, AND P. TSENG, *Hankel matrix rank minimization with applications to system identification and realization*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 946–977.

[10] T. D. HOCKING, A. JOULIN, F. BACH, AND J.-P. VERT, *Clusterpath an algorithm for clustering using convex fusion penalties*, in Proceedings of the International Conference on Machine Learning, 2011, pp. 745–752.

[11] L. JACOB, G. OBOZINSKI, AND J.-P. VERT, *Group lasso with overlap and graph lasso*, in Proceedings of the International Conference on Machine Learning, 2009, pp. 433–440.

[12] T. JIANG, *Sum-of-Norms Clustering: Theoretical Guarantee and Post-Processing*, Master's thesis, University of Waterloo, 2020.

[13] T. JIANG, S. VAVASIS, AND C. W. ZHAI, *Recovery of a mixture of Gaussians by sum-of-norms clustering*, J. Mach. Learn. Res., 21 (2020), pp. 1–16.

[14] N. KESKAR, J. NOCEDAL, F. ÖZTOPRAK, AND A. WAECHTER, *A second-order method for convex* 1*-regularized optimization with active-set prediction*, Optim. Methods Softw., 31 (2016), pp. 605–621.

[15] Y. LECUN, C. CORTES, AND C. J. C. BURGES, *The MNIST Database of Handwritten Digits*, 1998, http://yann.lecun.com/exdb/mnist/.

[16] M. X. LIN, Y. C. YUAN, D. F. SUN, AND K.-C. TOH, *Adaptive Sieving with PPDNA: Generating Solution Paths of Exclusive Lasso Models*, preprint, arXiv:2009.08719 [math.oc], 2020.

[17] F. LINDSTEN, H. OHLSSON, AND L. LJUNG, *Clustering using sum-of-norms regularization: With application to particle filter output computation*, in Proceedings of the IEEE Statistical Signal Processing Workshop, IEEE, 2011, pp. 201–204.

[18] S. LLOYD, *Least squares quantization in PCM*, IEEE Trans. Inform. Theory, 28 (1982), pp. 129–137.

[19] D. G. Mixon, S. Villar, and R. Ward, *Clustering subgaussian mixtures by semidefinite programming*, Inf. Inference, 6 (2017), pp. 389–415.
[20] J.-J. Moreau, *Proximité et dualité dans un espace Hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.
[21] S. A. Nene, S. K. Nayar, and H. Murase, *Columbia Object Image Library (COIL-100)*, Technical report CUCS-006-96, 1996.
[22] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547.
[23] A. Ng, M. Jordan, and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, in Proceedings of the Conference on Neural Information Processing Systems, 2001.
[24] A. Panahi, D. Dubhashi, F. D. Johansson, and C. Bhattacharyya, *Clustering by sum of norms: Stochastic incremental algorithm, convergence and cluster recovery*, in Proceedings of the International Conference on Machine Learning, 2017, pp. 2769–2777.
[25] K. Pelckmans, J. De Brabanter, J. A. Suykens, and B. De Moor, *Convex clustering shrinkage*, in Proceedings of the PASCAL Workshop on Statistics and Optimization of Clustering, 2005.
[26] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[27] Y. She, *Sparse regression with exact clustering*, Electron. J. Stat., 4 (2010), pp. 1055–1096.
[28] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., 22 (2000), pp. 888–905.
[29] M. Soltanolkotabi, E. Elhamifar, and E. J. Candes, *Robust subspace clustering*, Ann. Statist., 42 (2014), pp. 669–699.
[30] D. F. Sun, K.-C. Toh, and Y. C. Yuan, *Convex clustering: Model, theoretical guarantee and efficient algorithm*, J. Mach. Learn. Res., 22 (2021), pp. 1–32.
[31] K. M. Tan and D. Witten, *Statistical properties of convex clustering*, Electron. J. Stat., 9 (2015), pp. 2324–2347.
[32] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B Stat. Methodol., 58 (1996), pp. 267–288.
[33] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani, *Strong rules for discarding predictors in lasso-type problems*, J. R. Stat. Soc. Ser. B Stat. Methodol., 74 (2012), pp. 245–266.
[34] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, *Sparsity and smoothness via the fused lasso*, J. R. Stat. Soc. Ser. B Stat. Methodol., 67 (2005), pp. 91–108.
[35] R. Vidal, *Subspace clustering*, IEEE Signal Process. Mag., 28 (2011), pp. 52–68.
[36] J. Wang, P. Wonka, and J. Ye, *Lasso screening rules via dual polytope projection*, J. Mach. Learn. Res., 16 (2015), pp. 1063–1101.
[37] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation*, SIAM J. Sci. Comput., 32 (2010), pp. 1832–1857.
[38] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, J. R. Stat. Soc. Ser. B Stat. Methodol., 68 (2006), pp. 49–67.
[39] Y. C. Yuan, D. F. Sun, and K.-C. Toh, *An efficient semismooth Newton based algorithm for convex clustering*, in Proceedings of the International Conference on Machine Learning, 2018, pp. 5718–5726.
[40] E. H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory: Part* I. *Projections on convex sets: Part* II. *Spectral theory*, in Contributions to Nonlinear Functional Analysis, Elsevier, New York, 1971, pp. 237–424.
[41] Y. Zhou, R. Jin, and S. C.-H. Hoi, *Exclusive lasso for multi-task feature selection*, in Proceedings of the International Conference on Artificial Intelligence and Statistics, 2010, pp. 988–995.
[42] C. Zhu, H. Xu, C. Leng, and S. Yan, *Convex optimization procedure for clustering: Theoretical revisit*, in Proceedings of the Conference on Neural Information Processing Systems, 2014, pp. 1619–1627.