Mathematical Programming Computation (2025) 17:585–616
https://doi.org/10.1007/s12532-025-00282-2

FULL LENGTH PAPER

Adaptive sieving: a dimension reduction technique for sparse optimization problems

Yancheng Yuan[1] · Meixia Lin[2] · Defeng Sun[1] · Kim-Chuan Toh[3]

Received: 29 June 2023 / Accepted: 19 March 2025 / Published online: 28 April 2025

Abstract
In this paper, we propose an adaptive sieving (AS) strategy for solving general sparse machine learning models by effectively exploring the intrinsic sparsity of the solutions, wherein only a sequence of reduced problems with much smaller sizes need to be solved. We further apply the proposed AS strategy to generate solution paths for large-scale sparse optimization problems efficiently. We establish the theoretical guarantees for the proposed AS strategy including its finite termination property. Extensive numerical experiments are presented in this paper to demonstrate the effectiveness and flexibility of the AS strategy to solve large-scale machine learning models.

Keywords Adaptive sieving · Dimension reduction · Sparse optimization problems

Mathematics Subject Classification 90C06 · 90C25 · 90C90

✉ Meixia Lin
meixia_lin@sutd.edu.sg

Yancheng Yuan
yancheng.yuan@polyu.edu.hk

Defeng Sun
defeng.sun@polyu.edu.hk

Kim-Chuan Toh
mattohkc@nus.edu.sg

[1] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[2] Engineering Systems and Design, Singapore University of Technology and Design, Singapore, Singapore

[3] Department of Mathematics and Institute of Operations Research and Analytics, National University of Singapore, Singapore, Singapore

1 Introduction

Consider the convex composite optimization problems of the following form:

$$\min_{x\in\mathbb{R}^n} \Big\{\Phi(x) + P(x)\Big\}, \tag{1}$$

where $\Phi : \mathbb{R}^n \to \mathbb{R}$ is a convex twice continuously differentiable function and $P : \mathbb{R}^n \to (-\infty, +\infty]$ is a closed and proper convex function. The optimization problems in this form cover a wide class of models in modern data science applications and statistical learning. In practice, the regularizer $P(\cdot)$ is usually chosen to enforce sparsity with desirable structure in the estimators obtained from the model, especially in high-dimensional cases. For example, the Lasso regularizer [30] is proposed to force element-wise sparsity in the predictors, and the group lasso regularizer [37] is proposed to impose group-wise sparsity. Moreover, many more complicated regularizers have also been proposed to study other structured sparsity, such as the sparse group lasso regularizer [9, 15], the Sorted L-One Penalized Estimation (SLOPE) [3] and the exclusive lasso regularizer [18, 41].

Currently, many popular first-order methods have been proposed to solve the problems in the form of (1), such as the accelerated proximal gradient (APG) method [2], the alternating direction method of multipliers (ADMM) [8, 13] and the block coordinate descent (BCD) method [14, 27, 32]. These algorithms are especially popular in machine learning and statistics in recent years. However, first-order methods are often not robust and generally are only able to deliver low accuracy solutions. Other than first-order methods, some second-order methods have been developed for solving convex composite optimization problems in the form of (1), including the coderivative-based generalized Newton method [17], the forward-backward quasi-Newton method [28], the proximal trust-region method [1], and second-order algorithms based on the augmented Lagrangian method [6, 20, 23, 29].

However, both the first-order and second-order algorithms for solving the problems in the form of (1) will face significant challenges when scaling to high-dimensional problems. Fortunately, the desired solutions to the model in practice are usually highly sparse, where the nonzero entries correspond to selected features. Inspired by this property, in this paper, we propose an adaptive sieving strategy for solving sparse optimization models in the form of (1), by sieving out a large proportion of inactive features to significantly reduce the dimension of the problems, which can then highly accelerate the computation. The proposed AS strategy does not depend on the specific form of the regularizer $P(\cdot)$, as long as the proximal mapping of $P(\cdot)$ can be computed in an efficient way. It is worth emphasizing that our AS strategy can be applied to any solver providing that it is able to solve the reduced problems to the required accuracy. Numerical experiments demonstrate that our AS strategy is very effective in reducing the problem dimensions and accelerating the computation for solving sparse optimization models.

We will then apply the proposed AS strategy to generate solution paths for large-scale machine learning models, which is necessary for model selection via cross-validation in practice. In particular, we are interested in solving the following problem

for a sequence of values of the parameter λ:

$$\min_{x\in\mathbb{R}^n} \left\{\Phi(x) + \lambda P(x)\right\}. \tag{P_λ}$$

Here the hyper-parameter $\lambda > 0$ is added to control the trade-off between the loss and the sparsity level of the solutions. To reduce the computation time of obtaining a solution path, especially for high-dimensional cases, various feature screening rules, which attempt to drop some inactive features based on prior analysis, have been proposed. Tibshirani et al. [31] proposed a strong screening rule (SSR) based on the "unit-slope" bound assumption for the Lasso model and the regression problem with the elastic net regularizer. This idea has been extended to the SLOPE model recently [19]. Compared to this unsafe screening rule where some active features may be screened out by mistake, safe screening rules have been extensively studied. The first safe screening rule was proposed by El Ghaoui et al. [12] for Lasso models. Later on, Wang et al. [33, 34] proposed a dual polytope projection based screening rule (DPP) and an enhanced version (EDPP) for Lasso and group lasso models via carefully analyzing the geometry of the corresponding dual problems. Other safe screening rules, like Sphere test [36], have been proposed via different strategies for estimating compact regions containing the optimal solutions to the dual problems. The safe screening rules will not exclude active features, but in many instances they only exclude a small subset of inactive features due to their conservative nature. In addition, the safe screening rules are usually not applicable to generate a solution path for a sequence of the hyperparameters with large gaps. Recently, Zeng et al. [39] combined the SSR and the EDPP to propose a hybrid safe-strong screening rule, which was implemented in an R package `biglasso` [38]. However, these screening rules are usually problem specific and they are difficult to be applied to the models with general regularizers, like the exclusive lasso regularizer, as they are highly dependent on the separability and positive homogeneity of the regularizers. Also, they implicitly require the reduced problems to be solved exactly.

Based on our proposed AS strategy for solving the problem (1), we design a path generation method for generating solution paths of the machine learning model (P_λ), wherein a sequence of reduced problems with much smaller sizes need to be solved. Our path generation method exhibits great improvement over the existing screening rules in three aspects. First, it applies to the models with general regularizers, including those are non-separable or not positively homogeneous. Second, it does not require the reduced problems to be solved exactly. Third, as we will see in the numerical experiments, it is more aggressive, and is able to sieve out more inactive features. Here, it is worth mentioning that, the aforementioned screening rules are only for generating the solution path and they cannot be applied directly for solving a single problem with a fixed parameter.

The remaining part of the paper is organized as follows. In Sect. 2, we propose the adaptive sieving strategy for solving the sparse optimization models in the form of (1), followed by its theoretical analysis including the finite termination property in Sect. 3. In Sect. 4, we will design a path generation method for general sparse optimization models based on the proposed AS strategy. Section 5 provides the numerical

performance of the proposed AS strategy for solving various machine learning models on both synthetic data and real data. Experiments are also conducted to demonstrate the superior performance of the path generation method based on the AS strategy compared to other popular screening rules. Finally, we conclude the paper.

Notations: Denote the set $[n] = \{1, 2, \ldots, n\}$. For any $I \subseteq [n]$, $\bar{I}$ denotes the complement of I in $[n]$. For any $z \in \mathbb{R}$, $\mathrm{sign}(z)$ denotes the sign function of z. For any $x \in \mathbb{R}^n$, denotes its p-norm as $\|x\|_p = (\sum_{i=1}^p |x_i|^p)^{1/p}$. For simplicity, we denote $\|\cdot\| = \|\cdot\|_2$. For any vector $x \in \mathbb{R}^n$ and any index set $I \subseteq [n]$, denote x_I as the subvector generated by the elements of x indexed by I. For any closed and proper convex function $q : \mathbb{R}^n \to (-\infty, \infty]$, the proximal mapping of $q(\cdot)$ is defined by

$$\mathrm{Prox}_q(x) := \arg\min_{y \in \mathbb{R}^n} \left\{ q(y) + \frac{1}{2}\|y - x\|^2 \right\}, \tag{2}$$

for any $x \in \mathbb{R}^n$. It is known that $\mathrm{Prox}_q(\cdot)$ is Lipschitz continuous with modulus 1 [26].

2 Dimension reduction via adaptive sieving

Throughout this paper, we make the following blanket assumption, which is satisfied for many popular machine learning models, as discussed in [42, Section 2.1].

Assumption 1 Assume that the solution set of (1), denoted as Ω, is nonempty and compact.

2.1 The adaptive sieving strategy

We propose an adaptive sieving strategy for solving sparse optimization models in the form of (1). An appealing property of this strategy is that it does not depend on the specific form of the regularizer and thus can be applied to sparse optimization problems with general regularization functions. More importantly, due to the adaptive nature of the AS strategy, it can sieve out a very large proportion of inactive features to significantly reduce the dimension of the problems and accelerate the computation by a large margin. This further shows that the AS strategy actually serves as a powerful dimension reduction technique for sparse optimization models.

Algorithm 1 An adaptive sieving strategy for solving (1)

1: **Input**: an initial index set $I^0 \subseteq [n]$, a given tolerance $\epsilon \geq 0$ and a given positive integer $k_{\max}$ (e.g., $k_{\max} = 500$).
2: **Output**: an approximate solution x^* to the problem (1) satisfying $\|R(x^*)\| \leq \epsilon$.
3: **1**. Find

$$x^0 \in \arg\min_{x \in \mathbb{R}^n} \left\{ \Phi(x) + P(x) - \langle \delta^0, x \rangle \mid x_{\overline{I^0}} = 0 \right\}, \tag{3}$$

where $\delta^0 \in \mathbb{R}^n$ is an error vector such that $\|\delta^0\| \leq \epsilon$ and $(\delta^0)_{\overline{I^0}} = 0$.

2. Compute $R(x^0)$ and set $s = 0$.
4: **while** $\|R(x^s)\| > \epsilon$ **do**
5: **3.1**. Create J^{s+1} as

$$J^{s+1} = \left\{ j \in \overline{I^s} \mid (R(x^s))_j \neq 0 \right\}. \tag{4}$$

(We prove in Theorem 1 that $J^{s+1} \neq \emptyset$.) Let k be a positive integer satisfying $k \leq \min\{|J^{s+1}|, k_{\max}\}$ and define

$$\widehat{J}^{s+1} = \left\{ j \in J^{s+1} \mid |(R(x^s))_j| \text{ is among the first } k \text{ largest values in } \{|(R(x^s))_i|\}_{i \in J^{s+1}} \right\}.$$

Update I^{s+1} as:

$$I^{s+1} \leftarrow I^s \cup \widehat{J}^{s+1}.$$

6: **3.2**. Solve the following constrained problem with the initialization x^s,

$$x^{s+1} \in \arg\min_{x \in \mathbb{R}^n} \left\{ \Phi(x) + P(x) - \langle \delta^{s+1}, x \rangle \mid x_{\overline{I^{s+1}}} = 0 \right\}, \tag{5}$$

where $\delta^{s+1} \in \mathbb{R}^n$ is an error vector such that $\|\delta^{s+1}\| \leq \epsilon$ and $(\delta^{s+1})_{\overline{I^{s+1}}} = 0$.
7: **3.3**: Compute $R(x^{s+1})$ and set $s \leftarrow s + 1$.
8: **end while**
9: **return**: Set $x^* = x^s$.

We give the details of the AS strategy in Algorithm 1. Here in order to measure the accuracy of the approximate solutions, we define the proximal residual function $R : \mathbb{R}^n \to \mathbb{R}^n$ associated with the problem (1) as

$$R(x) := x - \mathrm{Prox}_P(x - \nabla\Phi(x)), \quad x \in \mathbb{R}^n. \tag{6}$$

The KKT condition of (1) implies that $\bar{x} \in \Omega$ if and only if $R(\bar{x}) = 0$.

A few remarks of Algorithm 1 are in order. First, in Algorithm 1, the introduction of the error vectors δ^0, $\{\delta^{s+1}\}$ in (3) and (5) implies that the corresponding minimization problems can be solved inexactly. It is important to emphasize that the vectors are not a priori given but they are the errors incurred when the original problems (with $\delta^0 = 0$ in (3) and $\delta^{s+1} = 0$ in (5)) are solved inexactly. We will explain how the error vectors δ^0, $\{\delta^{s+1}\}$ in (3) and (5) can be obtained in Sect. 3. Second, the initial active feature index set I^0 is suggested to be chosen such that $|I^0| \ll n$, in consideration of the

computational cost. Third, the sizes of the reduced problems (3) and (5) are usually much smaller than n, which will be demonstrated in the numerical experiments.

We take the least squares linear regression problem as an example to show an efficient initialization of the index set I^0 via the correlation test, which is similar to the idea of the surely independent screening rule in [10]. Consider $\Phi(x) = \frac{1}{2}\|Ax - b\|^2$, where $A = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^{m\times n}$ is a given feature matrix, $b \in \mathbb{R}^m$ is a given response vector. One can choose $k\lceil\sqrt{n}\rceil$ initial active features based on the correlation test between each feature vector a_i and the response vector b. That is, one can compute $s_i := |\langle a_i, b\rangle|/(\|a_i\|\|b\|)$ for $i = 1, \ldots, n$, and choose the initial guess of I^0 as

$$I^0 = \left\{i \in [n] : s_i \text{ is among the first } k\lceil\sqrt{n}\rceil \text{ largest values in } s_1, \ldots, s_n\right\}.$$

In practice, we usually choose $k = 10$.

2.2 Examples of the regularizer

As we shall see in Algorithm 1, there is no restriction on the form of the regularizer $P(\cdot)$. In particular, it is not necessary to be separable or positively homogeneous, which is an important generalization compared to the existing screening rules [12, 31, 33, 34, 36, 38, 39]. The only requirement for $P(\cdot)$ is that its proximal mapping $\mathrm{Prox}_P(\cdot)$ can be computed in an efficient way. Fortunately, it is the case for almost all popular regularizers used in practice. Below, we list some examples of the regularizers for better illustration.

– Lasso regularizer [30]:

$$P(x) = \lambda\|x\|_1, \quad x \in \mathbb{R}^n,$$

where $\lambda > 0$ is a given parameter.

– Elastic net regularizer [43]:

$$P(x) = \lambda_1\|x\|_1 + \lambda_2\|x\|^2, \quad x \in \mathbb{R}^n,$$

where $\lambda_1, \lambda_2 > 0$ are given parameters.

– Sparse group lasso regularizer [9, 15]:

$$P(x) = \lambda_1\|x\|_1 + \lambda_2\sum_{l=1}^{g} w_l\|x_{G_l}\|, \quad x \in \mathbb{R}^n,$$

where $\lambda_1, \lambda_2 > 0$, $w_1, \ldots, w_g \geq 0$ are parameters, and $\{G_1, \ldots, G_g\}$ is a disjoint partition of the set $[n]$. For the limiting case when $\lambda_1 = 0$, we get the group lasso regularizer [37].

– Exclusive lasso regularizer [18, 41]:

$$P(x) = \lambda\sum_{l=1}^{g}\|w_{G_l} \circ x_{G_l}\|_1^2, \quad x \in \mathbb{R}^n,$$

where $\lambda > 0$ is a given parameter, $w \in \mathbb{R}^n_{++}$ is a weight vector and $\{G_1, \ldots, G_g\}$ is a disjoint partition of the index set $[n]$.

– Sorted L-One Penalized Estimation (SLOPE) [3]:

$$P(x) = \sum_{i=1}^{n} \lambda_i |x|_{(i)},$$

with parameters $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ and $\lambda_1 > 0$. For a given vector $x \in \mathbb{R}^n$, we denote $|x|$ to be the vector in $\mathbb{R}^n$ obtained from x by taking the absolute value of its components. We define $|x|_{(i)}$ to be the i-th largest component of $|x|$ such that $|x|_{(1)} \geq \cdots \geq |x|_{(n)}$.

Among the above examples, the screening rules of the Lasso regularizer and the group lasso regularizer have been intensively studied in [12, 31, 33, 34, 36, 38, 39]. Recently, a strong screening rule has been proposed for SLOPE [19]. However, no unified approach has been proposed for sparse optimization problems with general regularizers. Fortunately, our proposed AS strategy is applicable to all the above examples, which includes three different kinds of "challenging" regularizers. In particular, (1) the sparse group lasso regularizer is a combination of two regularizers; (2) the SLOPE is not separable; (3) the exclusive lasso regularizer is not positively homogeneous.

3 Theoretical analysis of the AS strategy

In this section, we provide the theoretical analysis of the proposed AS strategy for solving sparse optimization problems presented in Algorithm 1. An appealing advantage of our proposed AS strategy is that it allows the involved reduced problems to be solved inexactly due to the introduction of the error vectors δ^0, $\{\delta^{s+1}\}$ in (3) and (5). It is important to emphasize that the vectors are not a priori given but they are the errors incurred when the original problems (with $\delta^0 = 0$ in (3) and $\delta^{s+1} = 0$ in (5)) are solved inexactly. The following proposition explains how the error vectors δ^0, $\{\delta^{s+1}\}$ in (3) and (5) can be obtained.

Proposition 1 *Given any $s = 0, 1, \cdots$. The updating rule of x^s in Algorithm 1 can be interpreted in the procedure as follows. Let M_s be a linear map from $\mathbb{R}^{|I^s|}$ to $\mathbb{R}^n$ defined as*

$$(M_s z)_{I^s} = z, \quad (M_s z)_{\overline{I^s}} = 0, \quad z \in \mathbb{R}^{|I^s|},$$

and Φ^s, P^s be functions from $\mathbb{R}^{|I^s|}$ to $\mathbb{R}$ defined as $\Phi^s(z) := \Phi(M_s z)$, $P^s(z) := P(M_s z)$ for all $z \in \mathbb{R}^{|I^s|}$. Then $x^s \in \mathbb{R}^n$ can be computed as

$$(x^s)_{I^s} := \mathrm{Prox}_{P^s}(\hat{z} - \nabla \Phi^s(\hat{z})),$$

and $(x^s)_{\overline{I^s}} = 0$*, where* $\hat{z}$ *is an approximate solution to the problem*

$$\min_{z\in\mathbb{R}^{|I^s|}} \left\{\Phi^s(z) + P^s(z)\right\}, \tag{7}$$

which satisfies

$$\|\hat{\delta}\| \le \epsilon, \quad \hat{\delta} := \hat{z} - (x^s)_{I^s} + \nabla\Phi^s((x^s)_{I^s}) - \nabla\Phi^s(\hat{z}), \tag{8}$$

where ϵ *is the parameter given in Algorithm 1.*

If the function $\Phi(\cdot)$ *is L-smooth, that is, it is continuously differentiable and its gradient is Lipschitz continuous with constant L:*

$$\|\nabla\Phi(x) - \nabla\Phi(y)\| \le L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n,$$

then the condition (8) *can be achieved by*

$$\|\hat{z} - \mathrm{Prox}_{P^s}(\hat{z} - \nabla\Phi^s(\hat{z}))\| \le \frac{\epsilon}{1+L}.$$

Proof Let $\{z^i\}$ be a sequence that converges to a solution of the problem (7). For any $i = 1, 2, \cdots$, define $\varepsilon^i := z^i - \mathrm{Prox}_{P^s}(z^i - \nabla\Phi^s(z^i)) + \nabla\Phi^s(\mathrm{Prox}_{P^s}(z^i - \nabla\Phi^s(z^i))) - \nabla\Phi^s(z^i)$. By the continuous differentiability of $\Phi^s(\cdot)$ and [7, Lemma 4.5], we know that $\lim_{i\to\infty}\|\varepsilon^i\| = 0$, which implies the existence of $\hat{z}$ in (8).

Next we explain the reason why the updating rule of x^s in Algorithm 1 can be interpreted as the one stated in the proposition. Since $(x^s)_{I^s} := \mathrm{Prox}_{P^s}(\hat{z} - \nabla\Phi^s(\hat{z}))$, we have

$$\hat{z} - (x^s)_{I^s} - \nabla\Phi^s(\hat{z}) \in \partial P^s((x^s)_{I^s}).$$

According to the definition of $\hat{\delta}$ in (8), we have

$$\hat{\delta} \in \nabla\Phi^s((x^s)_{I^s}) + \partial P^s((x^s)_{I^s}),$$

which means that x^s is the exact solution to the problem

$$\min_{x\in\mathbb{R}^n} \left\{\Phi(x) + P(x) - \langle\delta^s, x\rangle \;\middle|\; x_{\overline{I^s}} = 0\right\},$$

with $\delta^s := M_s\hat{\delta}$. Moreover, we can see that $(\delta^s)_{\overline{I^s}} = (M_s\hat{\delta})_{\overline{I^s}} = 0$ and

$$\|\delta^s\| = \|\hat{\delta}\| = \|\hat{z} - (x^s)_{I^s} + \nabla\Phi^s((x^s)_{I^s}) - \nabla\Phi^s(\hat{z})\| \le \epsilon.$$

If $\Phi(\cdot)$ is L-smooth, we have that

$$\|\hat{\delta}\| = \|\hat{z} - (x^s)_{I^s} + \nabla\Phi^s((x^s)_{I^s}) - \nabla\Phi^s(\hat{z})\|$$

$$
\begin{aligned}
&\leq \|\hat{z} - (x^s)_{I^s}\| + \|\nabla \Phi^s((x^s)_{I^s}) - \nabla \Phi^s(\hat{z})\| \\
&= \|\hat{z} - (x^s)_{I^s}\| + \|M_s^T \nabla \Phi(M_s(x^s)_{I^s}) - M_s^T \nabla \Phi(M_s \hat{z})\| \\
&\leq \|\hat{z} - (x^s)_{I^s}\| + \|\nabla \Phi(M_s(x^s)_{I^s}) - \nabla \Phi(M_s \hat{z})\| \\
&\leq \|\hat{z} - (x^s)_{I^s}\| + L\|M_s(x^s)_{I^s} - M_s \hat{z}\| \\
&= (1 + L)\|\hat{z} - (x^s)_{I^s}\|,
\end{aligned}
$$

which means that (8) can be achieved by

$$
\|\hat{z} - \mathrm{Prox}_{P^s}(\hat{z} - \nabla \Phi^s(\hat{z}))\| \leq \frac{\epsilon}{1 + L}.
$$

This completes the proof. □

Next, we will show the convergence properties of Algorithm 1 in the following proposition, which states that the algorithm will terminate after a finite number of iterations.

Theorem 1 *The while loop in Algorithm 1 will terminate after a finite number of iterations.*

Proof We first prove that when $\|R(x^s)\| > \epsilon \geq 0$ for some $s \geq 0$, the index set J^{s+1} defined in (4) is nonempty. We prove this by contradiction. Suppose that $J^{s+1} = \emptyset$, which means

$$
(R(x^s))_{\overline{I^s}} = 0.
$$

According to the definition of $R(\cdot)$ in (6), we have

$$
R(x^s) - \nabla \Phi(x^s) \in \partial P(x^s - R(x^s)).
$$

Note that x^s satisfies

$$
x^s \in \arg\min_{x \in \mathbb{R}^n} \left\{ \Phi(x) + P(x) - \langle \delta^s, x \rangle \;\middle|\; x_{\overline{I^s}} = 0 \right\},
$$

where $\delta^s \in \mathbb{R}^n$ is an error vector such that $(\delta^s)_{\overline{I^s}} = 0$ and $\|\delta^s\| \leq \epsilon$. By the KKT condition of the above minimization problem, we know that there exists a multiplier $y \in \mathbb{R}^n$ with $y_{I^s} = 0$ such that

$$
\begin{cases}
0 \in \nabla \Phi(x^s) + \partial P(x^s) - \delta^s - y, \\
\left(x^s\right)_{\overline{I^s}} = 0,
\end{cases}
$$

which means

$$
\delta^s + y - \nabla \Phi(x^s) \in \partial P(x^s).
$$

By the maximal monotonicity of the operator ∂P, we have

$$\langle R(x^s) - \delta^s - y, -R(x^s)\rangle \geq 0,$$

which implies that

$$\|R(x^s)\|^2 \leq \langle \delta^s + y, R(x^s)\rangle = \langle (\delta^s)_{I^s}, (R(x^s))_{I^s}\rangle \leq \|\delta^s\|\|R(x^s)\| \leq \epsilon\|R(x^s)\|.$$

Thus, we get $\|R(x^s)\| \leq \epsilon$, which is a contradiction.

Therefore, we have that for any $s \geq 0$, $J^{s+1} \neq \emptyset$ as long as $\|R(x^s)\| > \epsilon$, which further implies that $\widehat{J}^{s+1} \neq \emptyset$. In other words, new indices will be added to the index set I^{s+1} as long as the KKT residual has not achieved at the required accuracy. Since the total number of features n is finite, the while loop in Algorithm 1 will terminate after a finite number of iterations. □

Although we only have the finite termination guarantee for the proposed AS strategy, the superior empirical performance of the AS strategy will be demonstrated later in Sect. 5 with extensive numerical experiments. In particular, the number of AS iterations is no more than 5 (less than 3 for most of the cases) for solving a single sparse optimization problem in our experiments on both synthetic and real datasets.

4 An efficient path generation method based on the AS strategy

When solving machine learning models in practice, we need to solve the model with a sequence of hyper-parameters and then select appropriate values for the hyper-parameters based on some methodologies, such as cross-validation. In this section, we are going to propose a path generation method based on the AS strategy to obtain solution paths of general sparse optimization models. Specifically, we will borrow the idea of the proposed AS strategy to solve the model (P_λ) for a sequence of λ in a highly efficient way.

Before presenting our path generation method, we first briefly discuss the ideas behind the two most popular screening rules for obtaining solution paths of various machine learning models, namely the strong screening rule (SSR) [31] and the dual polytope projection based screening rule (DPP) [33, 34]. For convenience, we take the Lasso model as an illustration, with $\Phi(x) = \frac{1}{2}\|Ax - b\|^2$ and $P(x) = \|x\|_1$ in (P_λ), where $A = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^{m\times n}$ is the feature matrix and $b \in \mathbb{R}^m$ is the response vector. Let $x^*(\lambda)$ be an optimal solution to (P_λ). The KKT condition implies that:

$$a_i^T\theta^*(\lambda) \in \begin{cases} \{\lambda\ \mathrm{sign}(x^*(\lambda))\} & \text{if } x^*(\lambda)_i \neq 0 \\ [-\lambda, \lambda] & \text{if } x^*(\lambda)_i = 0 \end{cases},$$

where $\theta^*(\lambda)$ is the optimal solution of the associated dual problem:

$$\max_{\theta\in\mathbb{R}^m} \left\{ \frac{1}{2}\|b\|^2 - \frac{1}{2}\|\theta - b\|^2 \ \mid\ |a_i^T\theta| \leq \lambda,\ i = 1, 2, \ldots, n \right\}. \tag{9}$$

The existing screening rules for the Lasso model are based on the fact that $x^*(\lambda)_i = 0$ if $|a_i^T\theta^*(\lambda)| < \lambda$. The difference is how to estimate $a_i^T\theta^*(\lambda)$ based on an optimal solution $x^*(\tilde{\lambda})$ of $(P_{\tilde{\lambda}})$ for some $\tilde{\lambda} > \lambda$, without solving the dual problem (9). The SSR [31] discards the i-th predictor if

$$|a_i^T\theta^*(\tilde{\lambda})| \leq 2\lambda - \tilde{\lambda},$$

by assuming the "unit slope" bound condition: $|a_i^T\theta^*(\lambda_1) - a_i^T\theta^*(\lambda_2)| \leq |\lambda_1 - \lambda_2|$, for all $\lambda_1, \lambda_2 > 0$. Since this assumption may fail, the SSR may screen out some active features by mistake. Also, the SSR only works for consecutive hyper-parameters with a small gap, since it requires $\lambda > \frac{\tilde{\lambda}}{2}$. Wang et al. [33, 34] proposed the DPP by carefully analysing the properties of the optimal solution to the dual problem (9). The key idea is, if we can estimate a region Θ_λ containing $\theta^*(\lambda)$, then

$$\sup_{\theta\in\Theta_\lambda} |a_i^T\theta| < \lambda \quad \Longrightarrow \quad x^*(\lambda)_i = 0.$$

They estimate the region Θ_λ by realizing that the optimal solution of (9) is the projection onto a polytope. As we can see, a tighter estimation of Θ_λ will induce a better safe screening rule.

The bottlenecks of applying the above screening rules to solve general machine learning models in the form of (1) are: (1) the subdifferential of a general regularizer may be much more complicated than the Lasso regularizer, whose subdifferential is separable; (2) a general regularizer may not be positively homogeneous, which means that the optimal solution to the dual problem may not be the projection onto some convex set [25, Theorem 13.2]; (3) These screening rules may only exclude a small portion of zeros in practice even if the solutions of the problem are highly sparse, thus we still need to solve large-scale problems after the screening; (4) they implicitly require the reduced problems to be solved exactly, which is unrealistic for many machine learning models in practice.

To overcome these challenges, we propose a path generation method based on the AS strategy for solving the problem (P_λ) presented in Algorithm 2. Here we denote the proximal residual function $R_\lambda : \mathbb{R}^n \to \mathbb{R}^n$ associated with (P_λ) as

$$R_\lambda(x) := x - \mathrm{Prox}_{\lambda P}(x - \nabla\Phi(x)), \quad x \in \mathbb{R}^n. \tag{10}$$

Assume that for any $\lambda > 0$, the solution set of (P_λ) is nonempty and compact.

The design of the above algorithm directly leads us to the following theorem regarding its convergence property. The proof of the theorem can be obtained directly due to Theorem 1.

Theorem 2 *The solution path $x^*(\lambda_0), x^*(\lambda_1), \ldots, x^*(\lambda_k)$ generated by Algorithm 2 forms a sequence of approximate solutions to the problems $(P_{\lambda_0}), (P_{\lambda_1}), \cdots, (P_{\lambda_k})$, in the sense that*

$$\|R_{\lambda_i}(x^*(\lambda_i))\| \leq \epsilon, \quad i = 0, 1, \ldots, k.$$

Algorithm 2 An AS-based path generation method for (P_λ)

1: **Input**: a sequence of hyper-parameters: $\lambda_0 > \lambda_1 > \cdots > \lambda_k > 0$, and given tolerances $\epsilon \geq 0$ and $\hat{\epsilon} \geq 0$.
2: **Output**: a solution path: $x^*(\lambda_0), x^*(\lambda_1), x^*(\lambda_2), \ldots, x^*(\lambda_k)$.
3: **1**. Apply Algorithm 1 to solve the problem (P_{λ_0}) with the tolerance ϵ, and then denote the output as $x^*(\lambda_0)$. Let

$$I^*(\lambda_0) := \{j \mid |x^*(\lambda_0)_j| > \hat{\epsilon},\ j = 1, \ldots, n\}.$$

4: **for** $i = 1, 2, \ldots, k$ **do**
5: **2.1**. Let $I^0(\lambda_i) = I^*(\lambda_{i-1})$.
6: **2.2**. Apply Algorithm 1 to solve the problem (P_{λ_i}) with the initial index set $I^0(\lambda_i)$, the initialization $x^*(\lambda_{i-1})$ and the tolerance ϵ. Denote the output as $x^*(\lambda_i)$.
7: **2.3**. Let

$$I^*(\lambda_i) := \{j \mid |x^*(\lambda_i)_j| > \hat{\epsilon},\ j = 1, \ldots, n\}.$$

8: **end for**

In practice, we usually choose $\hat{\epsilon} = 10^{-10}$. We will compare the numerical performance of the proposed AS strategy to the popular screening rules in Sect. 5, our experiment results will show that the AS strategy is much more efficient for solving large-scale sparse optimization problems.

5 Numerical Experiments

In this section, we will present extensive numerical results to demonstrate the performance of the AS strategy.[1] We will test the performance of the AS strategy on the sparse optimization problems of the following form:

$$\min_{x \in \mathbb{R}^n} \Big\{ h(Ax) + \lambda P(x) \Big\}, \tag{11}$$

where $A : \mathbb{R}^n \to \mathbb{R}^m$ is a given data matrix, $h : \mathbb{R}^m \to \mathbb{R}$ is a convex, twice continuously differentiable loss function, $P : \mathbb{R}^n \to (-\infty, +\infty]$ is a proper closed convex regularization function, and $\lambda > 0$ is the hyper-parameter. More specifically, we consider the two most commonly used loss functions for $h(\cdot)$:

(1) (linear regression) $h(y) = \sum_{i=1}^m (y_i - b_i)^2/2$, for some given vector $b \in \mathbb{R}^m$;
(2) (logistic regression) $h(y) = \sum_{i=1}^m \log(1 + \exp(-b_i y_i))$, for some given vector $b \in \{-1, 1\}^m$.

For the regularization function $P(\cdot)$, we consider four important regularizers in statistical learning models: Lasso, sparse group lasso, exclusive lasso, and SLOPE. Details of the regularizers can be found in Sect. 2.2. The algorithms used for solving each sparse optimization model and its corresponding reduced problems when applying the AS strategy will be specified later in the related subsections.

[1] The code is available at https://doi.org/10.5281/zenodo.15087424.

In our experiments, we measure the accuracy of the obtained solution by the relative KKT residual:

$$\eta_{\mathrm{KKT}} := \frac{\|x - \mathrm{Prox}_{\lambda P}(x - A^T \nabla h(Ax))\|}{1 + \|x\| + \|A^T \nabla h(Ax)\|}, \tag{12}$$

as suggested in [20]. Here η_{KKT} measures how close the optimality condition of (11) is to being met, which is particularly crucial in scenarios where the function values decrease slowly but significant progress towards satisfying the optimality condition is being made. We terminate the tested algorithm when $\eta_{\mathrm{KKT}} \leq \varepsilon$, where $\varepsilon > 0$ is a given tolerance, which is set to be 10^{-6} by default. We choose $\hat{\epsilon} = 10^{-10}$ in Algorithm 2. All our numerical results are obtained by running MATLAB(2022a version) on a Windows workstation (Intel Xeon E5-2680 @2.50GHz).

The numerical experiments will be organized as follows. First, we present the numerical performance of the AS strategy on the mentioned models on synthetic data sets in Sects. 5.1 and 5.2. To better demonstrate the effectiveness and flexibility of our AS strategy, we compare it with other existing approaches for the path generation in Sect. 5.3, and test its performance with its reduced problems solved by different algorithms in Sect. 5.4. Finally, we present the numerical results on nine real data sets in Sect. 5.5.

5.1 Performance of the AS strategy for linear regression on synthetic data

In this subsection, we conduct numerical experiments to demonstrate the performance of the AS strategy for linear regression on synthetic data.

We first describe the generation of the synthetic data sets. We consider the models with group structure (e.g., sparse group lasso and exclusive lasso) and without group structure (e.g., Lasso and SLOPE). For simplicity, we randomly generate data with groups and test all the models on these data. Motivated by [4], we generate the synthetic data using the model $b = Ax^* + \xi$, where x^* is the predefined true solution and $\xi \sim \mathcal{N}(0, I_m)$ is a random noise vector. Given the number of observations m, the number of groups g and the number of features p in each group, we generate each row of the matrix $A \in \mathbb{R}^{m \times gp}$ by independently sampling a vector from a multivariate normal distribution $\mathcal{N}(0, \Sigma)$, where Σ is a Toeplitz covariance matrix with entries $\Sigma_{ij} = 0.9^{|i-j|}$ for the features in the same group, and $\Sigma_{ij} = 0.3^{|i-j|}$ for the features in different groups. For the ground-truth x^*, we randomly generate r nonzero elements in each group with i.i.d values drawn from the uniform distribution on $[0, 10]$. In particular, we test the case where $r = \lfloor 0.1\% \times p \rfloor$. More experiments for the case where $r = \lfloor 1\% \times p \rfloor$ are given in Appendix A.3.

Here, we mainly focus on solving the regression models in high-dimensional settings. In our experiments, we set $m \in \{500, 1000, 2000\}$. For the number of features, for simplicity, we fix g to be 20, but vary the number of features p in each group from 5000 to 60000. That is, we vary the total number of features $n = gp$ from $1,00,000$ to $12,00,000$.

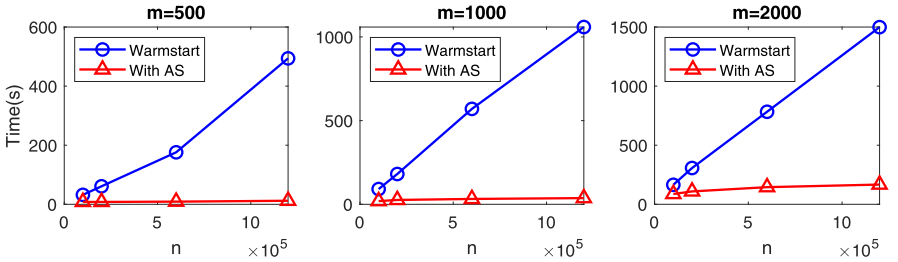


Fig. 1 Performance of the AS strategy applied to the generation of solution paths for the Lasso linear regression model on synthetic data sets with different problem sizes

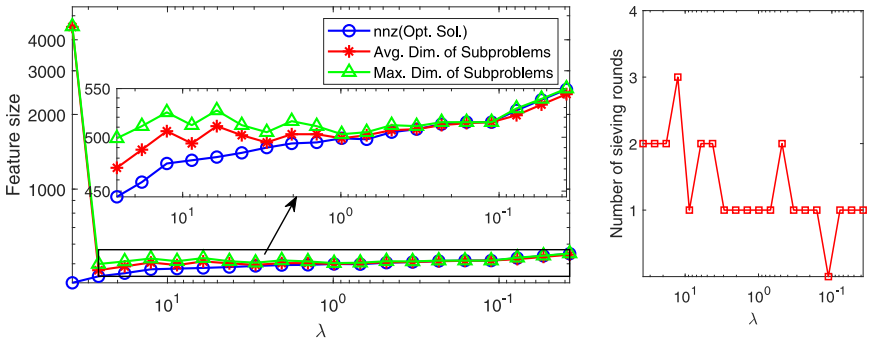
5.1.1 Numerical results on Lasso linear regression problems

In this subsection, we present the numerical results of the Lasso linear regression model. We will apply the state-of-the-art semismooth Newton based augmented Lagrangian (SSNAL) method[2] to solve the Lasso model and its corresponding reduced problems generated by the AS strategy. The comparison of the efficiency between the SSNAL method and other popular algorithms for solving Lasso models can be found in [20]. Following the numerical experiment settings in [20], we test the numerical performance of the AS strategy for generating a solution path for the Lasso linear regression model with $\lambda = \lambda_c \|A^T b\|_\infty$, where λ_c is taken from 10^{-1} to 10^{-4} with 20 equally divided grid points on the $\log_{10}$ scale.

We show the numerical results in Fig. 1, where the performance of the AS strategy and the well-known warm-start technique for generating solution paths of Lasso linear regression problems are compared with different problem sizes (m, n). From the numerical results, we can see that the AS strategy can significantly reduce the computational time of path generation. In particular, the AS strategy can accelerate the SSNAL method up to **43** times for generating solution paths for the Lasso linear regression model. Moreover, the AS strategy scales efficiently as the number of features increases, which makes it especially effective for handling high-dimensional data.

In order to further demonstrate the power of the AS strategy, in Fig. 2a, we plot the average (and maximum) problem size of the reduced problems in the AS strategy against the number of nonzero entries of the optimal solutions in the solution path for the case when $m = 500, n = 1,00,000$. We can see that the average reduced problem sizes in the AS strategy can nearly match the actual number of nonzeros in the optimal solutions, except for the first problem with the largest λ. This clearly demonstrates the dimensionality reduction achieved by the AS strategy when solving sparse optimization problems, especially in high-dimensional settings. In addition, Fig. 2(b) shows that we only need no more than three rounds of sieving in the AS strategy to solve the problem for each λ, which demonstrates that our sieving technique is quite effective in practice.

[2] https://github.com/MatOpt/SuiteLasso.



(a) Selected active features (b) Number of sieving rounds

Fig. 2 Performance profile of the AS strategy on the Lasso linear regression model for the case when $m = 500, n = 1,00,000$

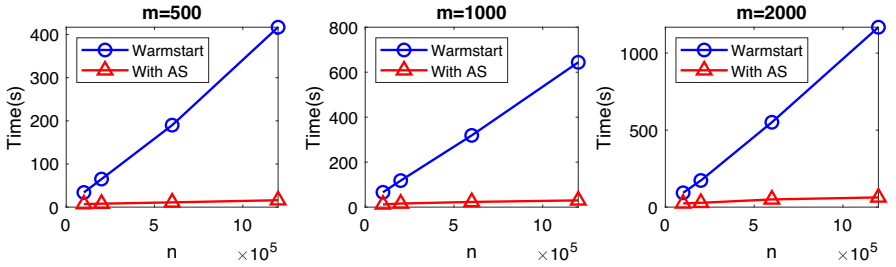
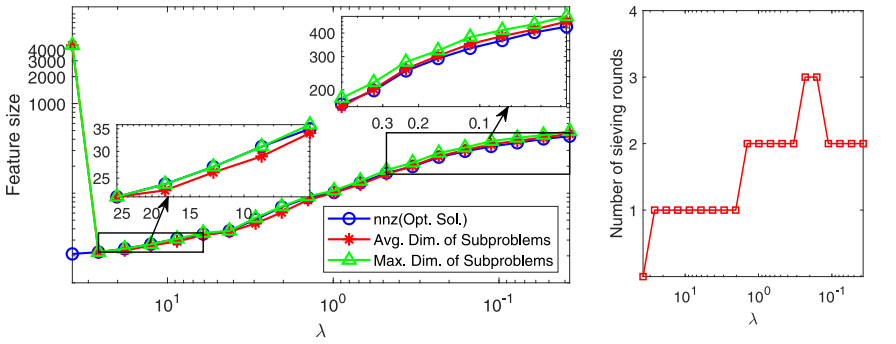


Fig. 3 Performance of the AS strategy applied to the generation of solution paths for the exclusive lasso linear regression model on synthetic data sets with different problem sizes

5.1.2 Numerical results on exclusive lasso linear regression problems

In this subsection, we present the numerical results of the exclusive lasso linear regression model. We apply the state-of-the-art dual Newton based proximal point algorithm (PPDNA) [21] to solve the exclusive lasso model and its corresponding reduced problems generated by the AS strategy. The comparison of the efficiency between the PPDNA and other popular algorithms for solving exclusive lasso models can be found in [21]. We follow the numerical experiment settings in [21] and test the numerical performance of generating a solution path for the exclusive lasso regression model with $\lambda = \lambda_c \|A^T b\|_\infty$ and w being the vector of all ones. Here, λ_c is taken from 10^{-1} to 10^{-4} with 20 equally divided grid points on the $\log_{10}$ scale.

We summarize the numerical results on the path generation of the exclusive lasso linear regression model with different problem sizes in Fig. 3. We can see that the AS strategy gives excellent performance in generating the solution paths for the exclusive lasso linear regression models, in the sense that it accelerates the PPDNA algorithm up to **30** times. In addition, the average (and maximum) problem sizes of the reduced problems against the number of nonzeros in the optimal solutions for the case when $m = 500, n = 1,00,000$ is shown in Fig. 4a. The number of sieving rounds for



(a) Selected active features (b) Number of sieving rounds

Fig. 4 Performance profile of the AS strategy on the exclusive lasso linear regression model for the case when $m = 500, n = 1,00,000$

this case is also provided in Fig. 4b. We can see that AS is able to highly reduce the dimensions of the optimization problems along the solution paths.

More experiments on the numerical performance of the AS strategy when solving the linear regression models with sparse group lasso regularizer and SLOPE regularzier are shown in Appendix A.1 and Appendix A.2, respectively.

5.2 Performance of the AS strategy for logistic regression on synthetic data

To test the regularized logistic regression problem, we generate A and x^* following the same settings as in Sect. 5.1 and define $b_i = 1$ if $Ax^* + \tilde{\xi} \geq 0$, and -1 otherwise, where $\tilde{\xi} \sim \mathcal{N}(0, I_m)$.

5.2.1 Numerical results on Lasso logistic regression problems

We present the results on the Lasso logistic regression model, where the (reduced) problems are again solved by the SSNAL method [20]. We test the performance on generating a solution path for the Lasso logistic regression model with $\lambda = \lambda_c \|A^T b\|_\infty$, where λ_c is taken from 10^{-1} to 10^{-4} with 20 equally divided grid points on the $\log_{10}$ scale. The detailed numerical results are shown in Table 1, where the performance of the AS strategy and the well-known warm-start technique for generating solution paths of Lasso logistic regression problems are compared on the problems with different sizes (m, n). For better illustration of the AS performance, we also list the total number of sieving rounds, together with the average (maximum) dimension of the reduced problems on the solution path.

We can see from the table that the AS strategy also works extremely well for solving the Lasso logistic regression problems, in the sense that it reduces the problem size by a large margin for each case and highly accelerates the computation of the path generation.

Table 1 Numerical performance of the AS strategy applied to the generation of solution paths (20 different λ's) for the Lasso logistic regression model on synthetic data sets. In the table, 'S. Rnd' represents the total number of the AS rounds during the path generation, 'Avg. D.' ('Max. D.') denotes the average (maximum) dimension of the reduced problems on the solution path

		Total time (hh:mm:ss)		Information of the AS		
m	$n(= g \times p)$	Warmstart	With AS	S. Rnd	Avg. D	Max. D
500	1, 00, 000	00:02:30	00:00:06	23	564	3170
	2, 00, 000	00:04:18	00:00:05	18	622	4473
	6, 00, 000	00:11:40	00:00:08	19	764	7752
	12, 00, 000	00:23:06	00:00:11	20	668	10,951
1000	100, 000	00:05:55	00:00:12	23	843	3227
	2, 00, 000	00:10:12	00:00:14	22	949	4512
	6, 00, 000	00:28:29	00:00:21	20	1119	7763
	12, 00, 000	00:53:27	00:00:28	18	1335	10,954
2000	1, 00, 000	00:17:27	00:01:12	26	1237	3251
	2, 00, 000	00:26:19	00:01:15	25	1553	4656
	6, 00, 000	01:09:52	00:01:23	22	1956	7859
	12, 00, 000	02:01:53	00:01:42	19	2282	11,010

5.2.2 Numerical results on exclusive lasso logistic regression problems

In this subsection, we present the numerical results on the exclusive lasso logistic regression model, where the (reduced) problems are solved by the PPDNA [21]. We test the numerical performance of the AS strategy for generating a solution path for the exclusive lasso logistic regression model with uniform weights and $\lambda = \lambda_c \|A^T b\|_\infty$, where λ_c is taken from 10^{-1} to 10^{-4} with 20 equally divided grid points on the $\log_{10}$ scale.

The results are presented in Table 2. As demonstrated, our AS strategy once again exhibits excellent performance in significantly reducing the problem sizes during the path generation of the exclusive lasso logistic regression problems. This reduction clearly leads to a noticeable acceleration in computation, further highlighting the efficiency and scalability of our approach.

5.3 Comparison between AS and other approaches

To further demonstrate the superior performance of the AS strategy, we will compare it with other popular approaches on the generation of solution paths for the Lasso linear regression models, which are implemented in high-quality publicly available packages. Specifically, we compare AS(+SSNAL) with the following approaches.

– Matlab's CD: The coordinate descend method [11], with its Matlab implementation in built-in "lasso" function in the Statistics and Machine Learning Toolbox.

Table 2 Numerical performance of the AS strategy applied to the generation of solution paths (20 different λ's) for the exclusive lasso logistic regression model on synthetic data sets

		Total time (hh:mm:ss)		Information of the AS		
m	$n(= g \times p)$	Warmstart	With AS	S. Rnd	Avg. D	Max. D
500	1, 00, 000	00:01:03	00:00:08	31	306	3161
	2, 00, 000	00:01:47	00:00:10	34	385	4471
	6, 00, 000	00:04:41	00:00:13	30	443	7751
	12, 00, 000	00:10:16	00:00:19	30	497	10,951
1000	1, 00, 000	00:01:36	00:00:21	36	429	3161
	2, 00, 000	00:03:03	00:00:27	34	566	4471
	6, 00, 000	00:07:26	00:00:35	36	623	7751
	1, 200, 000	00:14:44	00:00:48	34	702	10,951
2000	1, 00, 000	00:02:58	00:01:16	36	499	3161
	2, 00, 000	00:05:12	00:01:36	37	661	4471
	6, 00, 000	00:13:30	00:02:11	38	899	7751
	1, 200, 000	00:26:45	00:02:40	38	1038	10,951

– EDPP: The Enhanced Dual Projection onto Polytope (EDPP) rule implemented in the DPC Package,[3] where the inner problems are solved by the function "LeastR" in the solver SLEP [22].
– FPC_AS: The active-set based fixed point continuation method [35].

Note that the above approaches are terminated by different stopping criteria: AS is terminated if η_{KKT} in (12) is no greater than a given tolerance ε; Matlab's CD terminates when successive estimates of x differ in the ℓ_2 norm by a relative amount less than `RelTol`; EDPP is terminated if the relative successive estimates of x is no greater than `Tol`; and FPC_AS terminates if the maximum norm of sub-gradient is smaller than `gtol`.

For fair comparison, we test each approach with two choices of tolerances, and then compare their running time as well as the relative objective function values against the benchmark. Here we set the benchmark as $f^* := f(x^*) = \|Ax^* - b\|^2/2 + \lambda\|x^*\|_1$, where x^* is the solution obtained by the AS with a high accuracy ($\varepsilon = 10^{-8}$). The relateive objective function value associated with an approximate solution x is defined as $(f(x) - f^*)/f^*$. If this quantity is smaller, it means that the corresponding approximate solution is better.

The numerical comparison among the four approaches in generating solution paths for the Lasso model of size $(m, n) = (500, 1, 00, 000)$ with λ_c decreasing from 1 to 10^{-4} is shown in Table 3 and Fig. 5. It can be seen from the results that, the AS strategy gives great performance on path generation, as it achieves higher accuracy solutions in less time comparing with the existing approaches. In addition, from the experimental results, we can see that, as the tolerance becomes more stringent, our AS strategy exhibits strong scalability in terms of running time.

[3] http://dpc-screening.github.io/lasso.html.

Table 3 Running time of different approaches in generating solution paths for the Lasso model of size $(m, n) = (500, 1, 00, 000)$ with λ_c decreasing from 1 to 10^{-4}

AS		Matlab's CD		EDPP		FPC_AS	
ε	Time	RelTol	Time	Tol	Time	gtol	Time
1e−6	00:00:09	1e−3	00:00:09	1e−4	00:01:12	1e−6	00:01:06
1e−7	00:00:11	1e−4	00:00:27	1e−5	00:06:25	1e−7	00:04:28

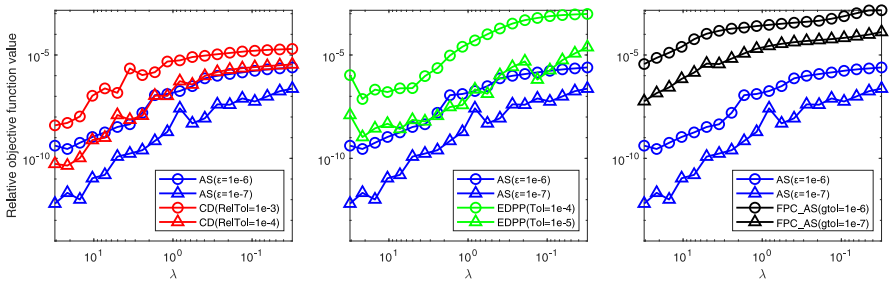


Fig. 5 Comparison among different approaches in generating solution paths for the Lasso model of size $(m, n) = (500, 1, 00, 000)$ with λ_c decreasing from 1 to 10^{-4}

5.4 Flexibility of the AS strategy

In this subsection, we combine our AS strategy with different algorithms for solving the reduced problems to test its generality and flexibility. The path generation for the Lasso model is taken as an example.

We incorporate the AS strategy with the SSNAL method, the alternating direction method of multipliers (ADMM) [8, 13], the accelerated proximal gradient (APG) method [2], and the coderivative-based generalized Newton method (GRNM) [17], separately. We stop the tested algorithm for solving each reduced problem if $\eta_{\mathrm{KKT}} \leq 10^{-6}$, the computation time exceeds 30 min, or the pre-set maximum number of iterations (100 for SSNAL, GRNM, and 20000 for ADMM, APG) is reached.

The numerical performance of the AS strategy combined with SSNAL, ADMM, APG and GRNM for solving the Lasso model of problem size $(m, n) = (500, 1, 00, 000)$ with λ_c decreasing from 1 to 10^{-4} is shown in Fig. 6. For illustration purpose, we also present the running time of each algorithm together with the warm-start techique for the path generation. As one can see from the figure, the AS strategy is highly efficient for generating solution paths of the Lasso models. Moreover, it can be incorporated with different algorithms for solving the reduced problems, and can consistently accelerate the computation. For example, when generating the solution path using the APG algorithm together with the warm-start technique, it takes a few hundred seconds for each λ, while with the AS technique, it takes less than one second. Additionally, the GRNM together with the warm-start technique is not able to solve the problems to the required accuracy within the specified time, while with the AS strategy, it is able to solve the problem for each λ within tens of seconds. The improvement comes from the fact that GRNM needs to solve an $n \times n$ linear system for the Newton step, which is

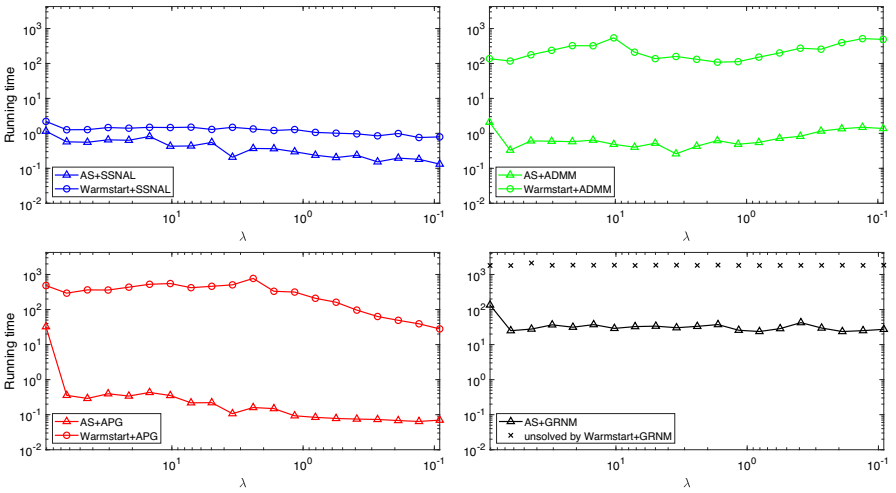


Fig. 6 The combination of the AS strategy with different algorithms for solving the reduced problems during the path generation for the Lasso model of size $(m, n) = (500, 1, 00, 000)$. The maker "×" in the plot means that the problem is not solved to the required accuracy

costly for high-dimensional problems. Moreover, since $n \gg m$, $A^T A$ becomes singular, leading to ill-conditioned (damped/regularized) Newton systems. In contrast, the AS strategy solves reduced, lower-dimensional problems, greatly speeding up GRNM and making it practical for high-dimensional tasks.

5.5 Performance of the AS strategy on real data sets

In this subsection, we present the numerical performance of the AS strategy on the real data sets for the aforementioned four linear regression models. We test the AS strategy for the Lasso model and the SLOPE model on eight UCI data sets, where the group information is not required. In addition, we test the AS strategy for the sparse group lasso model and the exclusive lasso model on one climate data set, where the group information is available.

Test instances from the UCI data repository. We test eight data sets from the UCI data repository [5]. The details are summarized in Table 4.

Following the settings in [20], we expand the original features by using polynomial basis functions over those features for the data sets pyrim, triazines, abalone, bodyfat, housing, mpg, and space-ga. For instance, the digit 5 in the name pyrim5 means that an order 5 polynomial is used to generate the basis functions.

NCEP/NCAR reanalysis 1 dataset. The data set [16] contains the monthly means of climate data measurements spread across the globe in a grid of $2.5^o \times 2.5^o$ resolutions (longitude and latitude 144×73) from 1948/1/1 to 2018/5/31. Each grid point (location) constitutes a group of 7 predictive variables (Air Temperature, Precipitable Water, Relative Humidity, Pressure, Sea Level Pressure, Horizontal Wind Speed and Vertical Wind Speed). Such data sets have two natural group structures: (i) groups over

Table 4 Details of the UCI test data. $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue

Name	(m, n)	$\lambda_{\max}(AA^T)$
abalone7	(4177, 6435)	5.21e+5
bcTCGA	(536, 17, 322)	1.13e+7
bodyfat7	(252, 1, 16, 280)	5.29e+4
housing7	(506, 77, 520)	3.28e+5
mpg7	(392, 3432)	1.28e+4
pyrim5	(74, 2, 01, 376)	1.22e+6
space-ga9	(3107, 5005)	4.01e+3
triazines4	(186, 6, 35, 376)	2.07e+7

Table 5 Running time of different approaches in generating solution paths for the Lasso model on mpg7 data set with λ_c decreasing from 1 to 10^{-4}

AS		Matlab's CD		EDPP		FPC_AS	
ε	Time	RelTol	Time	Tol	Time	gtol	Time
1e−6	00:00:06	1e−3	00:00:12	1e−4	00:00:04	1e−5	00:00:11
1e−7	00:00:06	1e−4	00:01:50	1e−5	00:00:22	1e−6	00:00:46

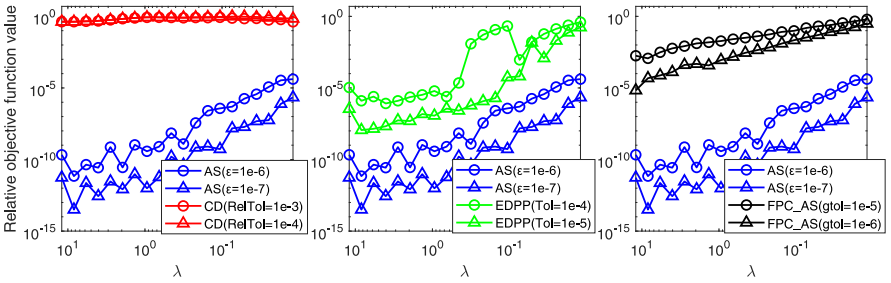


Fig. 7 Comparison among different approaches in generating solution paths for the Lasso model on mpg7 data set with λ_c decreasing from 1 to 10^{-4}

locations: 144×73 groups, where each group is of length 7; (ii) groups over features: 7 groups, where each group is of length 10,512. For both cases, the corresponding data matrix A is of dimension 845×73584.

5.5.1 Numerical results on Lasso linear regression model

In this subsection, we present the numerical results of the AS strategy for the Lasso linear regression model on the test instances from the UCI data repository.

First, we compare the AS strategy with Matlab's CD, EDPP, and FPC_AS on the path generation for the small data set mpg7. The running time of each approach with different choices of tolerances is shown in Table 5, and the relative objective function values along the solution path obtained by each approach are shown in Fig. 7. It can be

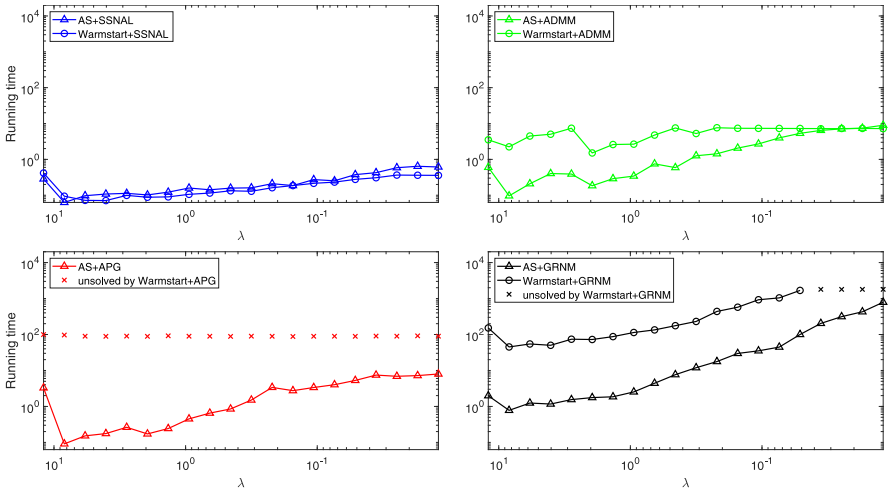


Fig. 8 The combination of the AS strategy with different algorithms for solving the reduced problems for the Lasso model on mpg7 data set

seen that the AS strategy outperforms other approaches by achieving more accurate solutions in less time. This is, it stands out in both speed and solution quality.

Next, we test the performance of the AS strategy combined with different algorithms for solving the reduced problems when solving the Lasso model on the small mpg7 data set. The results are shown in Fig. 8. We can see that the AS strategy can consistently improve the performance of SSNAL, ADMM, APG and GRNM, which again demonstrates its effectiveness and flexibility.

To better illustrate the generality and flexibility of the proposed AS strategy, in addition to the experiments on the small mpg7 data set, we conduct more experiments on other larger UCI data sets. For the algorithms to solve the corresponding models and their reduced problems, we take SSNAL and ADMM as two examples for illustration. We follow the same setting as described in Sect. 5.1.1 to choose the values for the parameter λ. The results are summarized in Table 6.

As one can see from Table 6, the SSNAL algorithm together with the warm-start technique already gives great performance, which takes less than 1 min to generate a solution path on each of the first seven data sets and takes around 8 min to generate a solution path on the last data set. Surprisingly, the SSNAL algorithm together with our proposed AS strategy can further speed up the path generation by a large margin, which can generate solution paths on all data sets within 45 s. In addition, the AS strategy also shows excellent performance when the reduced problems are solved by the ADMM algorithm. For example, for bodyfat7 data set, the ADMM algorithm with the warm-start technique takes more than 2 h to generate a solution path, while it only takes around 20 s with the AS strategy. Moreover, for the pyrim5 data set, the AS strategy allows the ADMM algorithm to generate a solution path with the required accuracy within 1 min, while it takes 3 h with the warm-start technique to generate a solution path with low accuracy.

Table 6 Numerical performance of the AS strategy applied to the generation of solution paths for Lasso linear regression models on UCI instances.

		Warmstart		With AS				
Data(Org.D.)	Alg	Time	η^*_{KKT}	Time	η^*_{KKT}	S.Rnd	Avg.D	Max.D
abalone7 (6435)	SSNAL	00:00:32	8.02e−7	00:00:22	9.82e−7	41	291	1505
	ADMM	01:16:54	1.00e−6	00:18:13	9.99e−7	41	293	1505
bcTCGA (17322)	SSNAL	00:00:11	9.82e−7	00:00:08	9.89e−7	38	547	1908
	ADMM	00:14:24	1.00e−6	00:00:11	9.76e−7	39	547	1908
bodyfat7 (116280)	SSNAL	00:00:23	9.05e−7	00:00:03	9.87e−7	39	574	6390
	ADMM	02:07:17	**8.44e–6**	00:00:21	9.94e−7	38	578	6391
housing7 (77520)	SSNAL	00:00:37	9.56e−7	00:00:13	9.78e−7	57	1129	12,634
	ADMM	02:30:10	**3.01e–5**	00:02:14	9.98e−7	57	1129	12,634
pyrim5 (201376)	SSNAL	00:00:57	9.75e−7	00:00:03	9.79e−7	41	492	4693
	ADMM	03:00:55	**1.45e–4**	00:00:53	9.92e−7	44	491	4693
space-ga9 (5005)	SSNAL	00:00:19	9.72e−7	00:00:12	9.93e−7	40	318	992
	ADMM	00:28:48	1.00e−6	00:08:48	9.99e−7	41	324	993
triazines4 (635376)	SSNAL	00:08:05	9.88e−7	00:00:44	9.90e−7	53	10,187	55,935
	ADMM	10:00:09	**9.10e–3**	02:46:28	**1.23e–5**	68	16,179	55,932

In the table, η^*_{KKT} denotes the worst relative KKT residual of the problems solved on the solution path, Org.D. denotes the original problem dimension, and the values in bold mean that the required accuracy not achieved

5.5.2 Numerical results on SLOPE linear regression model

In this subsection, we present the numerical results of the AS strategy for the SLOPE linear regression model on the test instances from the UCI data repository. For the weights $\lambda_{(i)}$ in the SLOPE model, we follow the experiment settings in [3]. We test the numerical performance on generating a solution path for the SLOPE regression model with $\lambda = \lambda_c \|A^T b\|_\infty$, where λ_c is taken from 10^{-1} to 10^{-4} with 20 equally divided grid points on the $\log_{10}$ scale. As for the algorithm to solve the corresponding models and their reduced problems, we take the state-of-the-art SSNAL method [24] and the popular first-order method ADMM for examples. It should be noted that, the AS strategy can be combined with any solver for solving the reduced problems. The results are summarized in Table 7.

From Table 7, we again observe the extremely good performance of the proposed AS strategy when combined with the SSNAL algorithm or the ADMM algorithm. For example, for the triazines4 data set, the SSNAL algorithm together with the warm-start technique takes more than 16 min to generate a solution path, while it only takes 42 s with the AS strategy. For the pyrim5 data set, the ADMM algorithm together with the warm-start technique takes more than one hour and 23 min to generate a solution path, while with the help of the AS strategy, it only takes 12 s for the path generation. From these experimental results, we can see that our proposed AS strategy can accelerate optimization algorithms for solving large-scale sparse optimization problems with intrinsic structured sparsity.

5.5.3 Numerical results on exclusive lasso linear regression model

We test the performance of the AS strategy for the exclusive lasso linear regression model on the NCEP/NCAR reanalysis 1 dataset. Since the exclusive lasso regularizer induces intra-group feature selections, we choose the group structure over features. In other words, there will be seven groups, and each group includes feature values of 10512 different locations. For the choice of the values for the parameter λ and the weight vector w, we follow the same setting as described in Sect. 5.1.2. To solve the reduced problems, we apply the PPDNA algorithm and the ADMM algorithm for illustration. The results are summarized in Table 8. We can observe that the AS strategy can accelerate the PPDNA algorithm by around 10 times and can accelerate the ADMM algorithm by more than 228 times. In addition, it also gives great performance in reducing the dimensions of the problems to be solved.

5.5.4 Numerical results on sparse group lasso linear regression model

We test the performance of the AS strategy for the sparse group lasso linear regression model on the NCEP/NCAR reanalysis 1 dataset. We follow the numerical experiment settings in [40], and we choose the group structure over locations. In other words, there will be 10512 groups, and each group includes 7 features. Following the settings in [40], we test the numerical performance of the AS strategy for generating a solution path for the sparse group lasso linear regression model with $w_l = \sqrt{|G_l|}$ for each $l = 1, \ldots, g$. For the parameters λ_1 and λ_2 in the sparse group lasso regularizer, we

Table 7 Numerical performance of the AS strategy applied to the generation of solution paths for SLOPE linear regression models on UCI instances

Data(Org.D.)	Alg	Warmstart		With AS				
		Time	η^*_{KKT}	Time	η^*_{KKT}	S.Rnd	Avg.D	Max.D
abalone7 (6435)	SSNAL	00:01:02	9.69e−7	00:00:31	4.12e−7	19	1010	4565
	ADMM	00:42:14	**6.35e–6**	00:09:34	9.58e−7	22	1049	4565
bcTCGA (17322)	SSNAL	00:00:33	6.09e−7	00:00:23	3.61e−7	24	571	899
	ADMM	00:02:44	9.99e−7	00:00:05	9.29e−7	24	571	899
bodyfat7 (116280)	SSNAL	00:01:24	8.81e−7	00:01:11	7.93e−7	14	19275	65,275
	ADMM	00:35:35	**1.16e–5**	00:02:54	9.06e−7	15	16871	41,866
housing7 (77520)	SSNAL	00:01:09	8.52e−7	00:00:23	5.84e−7	16	3036	11,191
	ADMM	00:18:29	**1.03e–6**	00:00:23	5.86e−7	16	2974	9299
mpg7 (3432)	SSNAL	00:00:06	6.70e−7	00:00:04	6.15e−7	21	321	982
	ADMM	00:00:25	1.00e−6	00:00:04	9.32e−7	22	310	544
pyrim5 (201376)	SSNAL	00:01:27	9.87e−7	00:00:10	4.04e−7	16	1948	12,191
	ADMM	01:23:55	**9.92e–4**	00:00:12	7.49e−7	16	1948	12,191
space-ga9 (5005)	SSNAL	00:00:31	8.54e−7	00:00:10	5.56e−7	18	246	1427
	ADMM	00:07:30	9.90e−7	00:01:56	8.50e−7	21	205	616
triazines4 (635376)	SSNAL	00:16:39	8.26e−7	00:00:42	4.64e−7	16	3779	29,465
	ADMM	08:52:56	**1.22e–4**	00:01:19	8.05e−7	16	3778	29,465

The values in bold mean that the required accuracy not achieved

Table 8 Numerical performance of the AS strategy applied to the generation of solution paths for the exclusive lasso linear regression models on the NCEP/NCAR reanalysis 1 dataset

		Warmstart		With AS				
Data(Org.D.)	Alg	Time	η^*_{KKT}	Time	η^*_{KKT}	S.Rnd	Avg.D	Max.D
NCEP/NCAR (73584)	PPDNA	00:01:18	9.85e−7	00:00:08	9.25e−7	14	178	2810
	ADMM	03:10:27	**3.96e–4**	00:00:50	9.39e−7	14	179	2810

The values in bold mean that the required accuracy not achieved

Table 9 Numerical performance of the AS strategy applied to the generation of solution paths for the sparse group lasso linear regression models on the NCEP/NCAR reanalysis 1 dataset

Data(Org.D.)	Alg	Warmstart		With AS				
		Time	η^*_{KKT}	Time	η^*_{KKT}	S.Rnd	Avg.D	Max.D
NCEP/NCAR (73584)	PPDNA	00:24:33	4.81e−7	00:00:27	9.81e−7	22	146	2711
	ADMM	02:48:46	**4.40e–3**	00:00:54	8.83e−7	24	145	2711

The values in bold mean that the required accuracy not achieved

take $\lambda_1 = \lambda_c \|A^T b\|_\infty$, $\lambda_2 = \lambda_1 / w_{\max}$ with $w_{\max} = \max\{w_1, \ldots, w_g\}$. Here, λ_c is taken from 10^{-1} to 10^{-3} with 20 equally divided grid points on the $\log_{10}$ scale. As for solving the reduced problems in the AS procedure, we employ the state-of-the-art SSNAL method [40]. The comparison of the efficiency and robustness between the SSNAL method and other popular algorithms for solving sparse group lasso models is presented in [40]. In addition, the popular first-order method ADMM algorithm is also employed.

The results are summarized in Table 9. It can be seen from the table that, the AS strategy can accelerate the SSNAL algorithm by around 54 times and accelerate the ADMM algorithm by around 187 times.

6 Conclusion

In this paper, we design an adaptive sieving strategy for solving general sparse optimization models and further generating solution paths for a given sequence of parameters. For each reduced problem involved in the AS strategy, we allow it to be solved inexactly. The finite termination property of the AS strategy has also been established. Extensive numerical experiments have been conducted to demonstrate the effectiveness and flexibility of the AS strategy to solve large-scale machine learning models.

A More numerical experiments on synthetic data

In this section, we present more experiments to demonstrate the performance of the AS strategy to generate solution paths for sparse optimization problems.

A.1 Numerical results on sparse group lasso linear regression problems

In this subsection, we present the numerical results on the sparse group lasso linear regression model. We apply the state-of-the-art SSNAL method [40] to solve the sparse group lasso model and its corresponding reduced problems generated by the AS strategy during the path generation. For the choice of the values for the parameters λ_1, λ_2 and w, we follow the same setting as described in Sect. 5.5.4, and λ_c is taken from 10^{-1} to 10^{-4} with 20 equally divided grid points on the $\log_{10}$ scale.

Table 10 Numerical performance of the AS strategy applied to the generation of solution paths for the sparse group lasso linear regression model on synthetic data sets with sparsity level 0.1%

		Total time (hh:mm:ss)		Information of the AS		
m	$n(= g \times p)$	Warmstart	With AS	S. Rnd	Avg. D	Max. D
500	1, 00, 000	00:03:00	00:00:35	17	1743	3588
	200, 000	00:04:58	00:00:38	16	2234	4829
	600, 000	00:14:11	00:00:55	21	2764	7839
	1, 200, 000	00:27:34	00:01:00	20	3215	10998
1000	1, 00, 000	00:05:47	00:02:36	22	2000	4259
	200, 000	00:10:28	00:02:16	19	2827	6295
	600, 000	00:31:02	00:02:16	17	3570	8311
	1, 200, 000	00:52:40	00:02:21	16	4286	11252
2000	1, 00, 000	00:15:46	00:06:19	34	2062	3464
	200, 000	00:23:18	00:06:46	33	3006	6321
	600, 000	00:58:03	00:05:31	25	4742	11063
	1, 200, 000	01:55:24	00:07:07	26	5327	12557

We summarize the numerical results of the comparison of the AS strategy and the warm-start technique for generating solution paths of sparse group lasso linear regression problems in Table 10. As we can see from the table, the AS strategy significantly accelerates the path generation of the sparse group lasso linear regression models. For example, on the synthetic data set of size $(m, n) = (2000, 1, 200, 000)$, generating a solution path of the sparse group lasso problem with the warm-start technique takes around 2 h, while with our proposed AS strategy, it only takes around 7 min. The superior performance of the AS strategy can also been seen from the much smaller sizes of the reduced problems compared with the original problem sizes.

A.2 Numerical results on SLOPE linear regression problems

Here, we present the results on SLOPE linear regression model. We will apply the state-of-the-art SSNAL method [24] to solve the SLOPE regression model and its corresponding reduced problems generated by the AS strategy. The comparison of the efficiency between the SSNAL method and other popular algorithms for solving SLOPE models can be found in [24]. The choice of the values for the parameter λ follows from the same setting as described in Sect. 5.5.2.

The results are presented in Table 11. We again observe the superior performance of the AS strategy on solving sparse optimization models. Specifically, when we apply the warm-start technique to generate solution paths for the SLOPE linear regression model with $(m, n) = (2000, 1, 200, 000)$, it runs out of memory, while with our AS strategy, it takes less than 12 min. Moreover, our AS strategy gives great performance in reducing the problem sizes during the path generation.

Table 11 Numerical performance of the AS strategy applied to the generation of solution paths for the SLOPE linear regression model on synthetic data sets with sparsity level 0.1%

		Total time (hh:mm:ss)		Information of the AS		
m	$n(= g \times p)$	Warmstart	With AS	S. Rnd	Avg. D	Max. D
500	1, 00, 000	00:01:38	00:00:38	22	1484	3122
	2, 00, 000	00:02:42	00:00:36	19	1962	4474
	6, 00, 000	00:06:30	00:00:37	17	2314	3922
	1, 200, 000	00:13:31	00:00:44	14	3140	10,954
1000	1, 00, 000	00:04:30	00:02:33	21	2007	3000
	2, 00, 000	00:05:46	00:02:09	19	2981	4701
	6, 00, 000	00:14:44	00:02:44	21	4476	7988
	1, 200, 000	00:27:17	00:02:38	17	5050	11,151
2000	1, 00, 000	00:10:24	00:05:39	21	2514	4914
	2, 00, 000	00:17:31	00:09:38	21	3571	5553
	6, 00, 000	00:31:42	00:10:29	22	7038	8619
	1, 200, 000	out-of-memory	00:11:26	23	8322	11,234

Table 12 Numerical performance of the AS strategy applied to the generation of solution paths for the Lasso linear regression model on synthetic data sets with sparsity level at 1%

		Total time (hh: mm: ss)		Information of the AS		
m	$n(= g \times p)$	Warmstart	With AS	S. Rnd	Avg. D	Max. D
500	1, 00, 000	00:00:53	00:00:07	31	682	3310
	2, 00, 000	00:01:14	00:00:07	26	765	4503
	6, 00, 000	00:04:49	00:00:09	25	830	7760
	1, 200, 000	00:08:46	00:00:12	23	997	10,956
1000	1, 00, 000	00:01:17	00:00:25	29	1304	4958
	2, 00, 000	00:02:28	00:00:27	29	1345	5180
	6, 00, 000	00:07:03	00:00:34	31	1554	7898
	1, 200, 000	00:12:55	00:00:38	27	1644	11,035
2000	100, 000	00:03:20	00:02:14	33	2372	11074
	2, 00, 000	00:05:04	00:02:17	34	2574	11,172
	6, 00, 000	00:13:27	00:02:22	32	2557	9357
	1, 200, 000	00:25:30	00:02:39	31	2773	11,757

A.3 Numerical results on data sets with other sparsity level

As mentioned in Sect. 5.1, in the data generation procedure, we set each group of the ground-truth x^* to contain r nonzero elements. We have tested the case when $r = \lfloor 0.1\% \times p \rfloor$ in Sects. 5.1–5.4, A.1 and A.2. In this subsection, we present more experimental results for the case when $r = \lfloor 1\% \times p \rfloor$.

Table 13 Numerical performance of the AS strategy applied to the generation of solution paths for the exclusive lasso linear regression model on synthetic data sets with sparsity level at 1%

		Total time (hh:mm:ss)		Information of the AS		
m	$n(= g \times p)$	Warmstart	With AS	S. Rnd	Avg. D	Max. D
500	1, 00, 000	00:00:34	00:00:05	29	199	3161
	2, 00, 000	00:01:09	00:00:05	30	206	4471
	6, 00, 000	00:03:06	00:00:08	24	284	7751
	1, 200, 000	00:07:03	00:00:14	22	343	10,951
1000	1, 00, 000	00:00:49	00:00:10	29	213	3161
	2, 00, 000	00:01:37	00:00:12	31	234	4471
	6, 00, 000	00:04:19	00:00:17	22	298	7751
	1, 200, 000	00:08:18	00:00:23	21	348	10,951
2000	1, 00, 000	00:01:35	00:00:26	25	196	3161
	2, 00, 000	00:02:54	00:00:29	25	241	4471
	6, 00, 000	00:08:04	00:00:35	23	302	7751
	1, 200, 000	00:16:02	00:00:50	22	356	10,951

The numerical performance of the AS strategy applied to the generation of solution paths for the Lasso linear regression model on synthetic data sets with sparsity level at 1% is shown in Table 12, and that for the exclusive lasso linear regression model is shown in Table 13. As demonstrated, the AS strategy once again showcases its superior performance for solving sparse optimization models by greatly accelerating the path generation and significantly reducing the problem dimensions.

Acknowledgements The authors wish to thank the associate editor and three anonymous reviewers, whose insightful comments helped to improve the paper.

Funding Yancheng Yuan is supported in part by The Hong Kong Polytechnic University under Grants P0038284/P0045485, and the Research Center for Intelligent Operations Research. Meixia Lin is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 grant call (MOE-T2EP20123-0013). Defeng Sun is supported in part by the Hong Kong Research Grant Council under Grant 15304721. Kim-Chuan Toh is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 3 grant call (MOE-2019-T3-1-010).

Data availability All data analyzed during this study are publicly available. References are included in this published article.

Code availability The full code is made publicly available. We remark that a set of packages were used in this study, that were either open source or available for academic use. Specific references are included in this published article.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Baraldi, R.J., Kouri, D.P.: A proximal trust-region method for nonsmooth optimization with inexact function and gradient evaluations. Math. Program. **201**, 559–598 (2023)
2. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)
3. Bogdan, M., Van Den Berg, E., Sabatti, C., Su, W., Candès, E.J.: SLOPE-adaptive variable selection via convex optimization. Ann. Appl. Stat. **9**, 1103 (2015)
4. Campbell, F., Allen, G.I.: Within group variable selection through the exclusive lasso. Electron. J. Stat. **11**, 4220–4257 (2017)
5. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 1–27 (2011)
6. Cui, Y., Pang, J.-S., Sen, B.: Composite difference-max programs for modern statistical estimation problems. SIAM J. Optim. **28**, 3344–3374 (2018)
7. Du, M.: An Inexact Alternating Direction Method of Multipliers for Convex Composite Conic Programming with Nonlinear Constraints, PhD thesis, Department of Mathematics, National University of Singapore, Singapore (2015)
8. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**, 293–318 (1992)
9. Eldar, Y.C., Mishali, M.: Robust recovery of signals from a structured union of subspaces. IEEE Trans. Inf. Theory **55**, 5302–5316 (2009)
10. Fan, J., Lv, J.: Sure independence screening for ultrahigh dimensional feature space. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **70**, 849–911 (2008)
11. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**, 1 (2010)
12. Ghaoui, L.E., Viallon, V., Rabbani, T.: Safe feature elimination for the lasso and sparse supervised learning problems. Pac. J. Optim. **8**, 667–698 (2012)
13. Glowinski, R., Marroco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. Revue Fr. d'automatique Inf. Rech. Op. Anal. Numér. **9**, 41–76 (1975)
14. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. Oper. Res. Lett. **26**, 127–136 (2000)
15. Jacob, L., Obozinski, G., Vert, J.-P.: Group lasso with overlap and graph lasso. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 433–440 (2009)
16. Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., Zhu, Y., Chelliah, M., Ebisuzaki, W., Higgins, W., Janowiak, J., Mo, K.C., Ropelewski, C., Wang, J., Leetmaa, A., Reynolds, R., Jenne, R., Joseph, D.: The NCEP/NCAR 40-year reanalysis project. Bull. Am. Meteor. Soc. **77**, 437–472 (1996)
17. Khanh, P.D., Mordukhovich, B.S., Phat, V.T., Tran, D.B.: Globally convergent coderivative-based generalized Newton methods in nonsmooth optimization. Math. Program. **205**, 373–429 (2024)
18. Kowalski, M.: Sparse regression using mixed norms. Appl. Comput. Harmon. Anal. **27**, 303–324 (2009)
19. Larsson, J., Bogdan, M., Wallin, J.: The strong screening rule for SLOPE. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20 (2020)
20. Li, X., Sun, D.F., Toh, K.-C.: A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. SIAM J. Optim. **28**, 433–458 (2018)
21. Lin, M., Yuan, Y., Sun, D., Toh, K.-C.: A highly efficient algorithm for solving exclusive lasso problems. Optim. Methods Softw. **39**, 489–518 (2024)
22. Liu, J., Ji, S., Ye, J.: SLEP: sparse learning with efficient projections. Arizona State Univ. **6**, 7 (2009)
23. Liu, R., Pan, S., Wu, Y., Yang, X.: An inexact regularized proximal Newton method for nonconvex and nonsmooth optimization. Comput. Optim. Appl. **88**, 603–641 (2024)
24. Luo, Z., Sun, D.F., Toh, K.-C., Xiu, N.: Solving the OSCAR and SLOPE models using a semismooth Newton-based augmented Lagrangian method. J. Mach. Learn. Res. **20**, 1–25 (2019)
25. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
26. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control. Optim. **14**, 877–898 (1976)

27. Sardy, S., Bruce, A.G., Tseng, P.: Block coordinate relaxation methods for nonparametric wavelet denoising. J. Comput. Graph. Stat. **9**, 361–379 (2000)
28. Stella, L., Themelis, A., Patrinos, P.: Forward-backward quasi-Newton methods for nonsmooth optimization problems. Comput. Optim. Appl. **67**, 443–487 (2017)
29. Tang, P., Wang, C., Jiang, B.: A proximal-proximal majorization-minimization algorithm for nonconvex rank regression problems. IEEE Trans. Signal Process. **71**, 3502–3517 (2023)
30. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. Ser. B (Methodol.) **58**, 267–288 (1996)
31. Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., Tibshirani, R.J.: Strong rules for discarding predictors in lasso-type problems. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **74**, 245–266 (2012)
32. Tseng, P.: Dual coordinate ascent methods for non-strictly convex minimization. Math. Program. **59**, 231–247 (1993)
33. Wang, J., Wonka, P., Ye, J.: Lasso screening rules via dual polytope projection. J. Mach. Learn. Res. **16**, 1063–1101 (2015)
34. Wang, J., Zhou, J., Wonka, P., Ye, J.: Lasso screening rules via dual polytope projection. In: Advances in Neural Information Processing Systems, pp. 1070–1078 (2013)
35. Wen, Z., Yin, W., Goldfarb, D., Zhang, Y.: A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. SIAM J. Sci. Comput. **32**, 1832–1857 (2010)
36. Xiang, Z.J., Wang, Y., Ramadge, P.J.: Screening tests for lasso problems. IEEE Trans. Pattern Anal. Mach. Intell. **39**, 1008–1027 (2016)
37. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **68**, 49–67 (2006)
38. Zeng, Y., Breheny, P.: The biglasso package: a memory- and computation-efficient solver for Lasso model fitting with big data in R. R J. **12**, 6–19 (2021)
39. Zeng, Y., Yang, T., Breheny, P.: Hybrid safe-strong rules for efficient optimization in lasso-type problems. Comput. Stat. Data Anal. **153**, 107063 (2021)
40. Zhang, Y., Zhang, N., Sun, D.F., Toh, K.-C.: An efficient Hessian based algorithm for solving large-scale sparse group Lasso problems. Math. Program. **179**, 223–263 (2020)
41. Zhou, Y., Jin, R., Hoi, S.C.-H.: Exclusive lasso for multi-task feature selection. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 988–995 (2010)
42. Zhou, Z., So, A.M.-C.: A unified approach to error bounds for structured convex optimization problems. Math. Program. **165**, 689–728 (2017)
43. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **67**, 301–320 (2005)