Analytical and Numerical Solutions for Nonlinear Equations
©Available online at https://ansne.du.ac.ir/
Online ISSN: 3060–785X
2026, Volume 11, Issue 2, pp. 273–288
Research article

An Efficient LASSO Estimation for Time-Dependent Cox Models via Adaptive Active-Set Coordinate Descent

Rohollah Ramezani[1], Mohammad Reza Rabiei[1,*], Mohammad Arashi[2]

[1] Department of Statistics, Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran

[2] Department of Statistics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, P.O. Box 1159, Mashhad 91775, Iran

* **Corresponding author(s):** rabiei_stat@shahroodut.ac.ir

Received: 17/04/2026 **Revised:** 24/05/2026 **Accepted:** 18/06/2026 **Published:** 04/09/2026 10.22128/ansne.2026.3301.1211

Abstract

The penalized Cox proportional hazard model is a popular analytical approach for survival data with a large set of covariates. Such problems are especially challenging when covariates vary over follow-up time (i.e., the covariates are time-dependent), leading to increased computational complexity and difficulties in efficient variable selection. In this paper, we propose an Adaptive Active-Set Coordinate Descent (AACD) algorithm for LASSO-penalized time-dependent Cox models. The proposed method combines coordinate descent with an adaptive active-set strategy and warm starts along the regularization path, allowing the algorithm to focus computation on relevant variables and better exploit sparsity. Simulation studies demonstrate that AACD consistently outperforms `glmnet`, achieving uniformly lower mean squared errors with relative efficiency greater than one across all settings, and reducing estimation error by up to 37% in small samples. The method remains robust as dimensionality increases, improves variable selection by reducing false negatives, and produces more parsimonious models with comparable predictive performance while reducing runtime by approximately 36% in real data analysis.

Keywords: Adaptive active-set, Coordinate descent, LASSO, Time-dependent cox model, Variable selection.

Mathematics Subject Classification (2020): 62N01, 62N02, 62F15

1 Introduction

The Cox proportional hazards model [1, 4] is a fundamental framework in survival analysis for modeling time-to-event data. In many applications, covariates vary over follow-up time, leading to time-dependent Cox models that provide greater modeling flexibility. However, such settings introduce additional computational complexity, particularly as the number of covariates increases. In these situations, classical estimation procedures can become computationally intensive or unstable, making efficient variable selection essential [2, 8].

Penalized regression methods provide a practical solution for simultaneous estimation and variable selection. Among these, the least absolute shrinkage and selection operator (LASSO) [16] is widely used for producing sparse solutions while remaining computationally tractable. Efficient software implementations, such as `glmnet`, have facilitated its application in practice [12]. Related penalization strategies include the elastic net [20] and nonconvex penalties such as SCAD [5]. Several algorithms have been developed to efficiently

COPYRIGHTS: ©2026, Analytical and Numerical Solutions for Nonlinear Equations published by Damghan University.
This is an open access article published under the *CC BY 4 DEED* license

solve the optimization problems arising in penalized Cox models [7, 13, 19].

Despite these developments, efficient optimization methods for LASSO-penalized Cox models with time-dependent covariates remain an active area of research. Strategies that focus computation on a subset of relevant variables, such as screening rules or active-set approaches, can substantially reduce computational cost and improve numerical stability [11, 17]. These strategies become particularly valuable as the number of covariates increases, especially in the presence of time-dependent structures.

In this paper, we propose an adaptive active-set coordinate descent (AACD) algorithm for fitting LASSO-penalized time-dependent Cox models. The proposed method integrates coordinate descent, warm starts along the regularization path, and an adaptive active-set strategy that focuses updates on variables most likely to affect the objective function. This design substantially improves computational efficiency while preserving estimation accuracy as dimensionality increases.

Contributions. The main contributions of this work are threefold. First, we develop the AACD algorithm for efficiently computing the regularization path in LASSO-penalized time-dependent Cox models. Second, we introduce an adaptive active-set mechanism that dynamically restricts updates to relevant variables, thereby reducing unnecessary computations. Third, through extensive simulation studies, we demonstrate the advantages of AACD in terms of estimation accuracy, variable selection, and computational efficiency as dimensionality increases. Overall, AACD provides a flexible and computationally efficient framework for survival analysis as dimensionality increases.

2 The LASSO-Penalized TimeDependent Cox Model

Consider right-censored survival data. Let T_i and C_i denote the event and censoring times for subject i, respectively. The observed time is $t_i = \min(T_i, C_i)$ with event indicator $\delta_i = I(T_i \le C_i)$ for $i = 1, \ldots, n$. Let $\mathbf{X}_i(t) = (X_{i1}(t), \ldots, X_{ip}(t))^\top$, $t \in [0, t_i]$, represent a p-dimensional vector of possibly time-dependent covariates.

The time-dependent Cox model specifies the conditional hazard function as

$$h_i(t \mid \mathbf{X}_i(t)) = h_0(t) \exp\left(\beta^\top \mathbf{X}_i(t)\right), \tag{1}$$

where $h_0(t)$ is an unspecified baseline hazard and $\beta = (\beta_1, \ldots, \beta_p)^\top$ are the regression coefficients.

Inference for β relies on the partial likelihood. Let $R(t) = \{j : t_j \ge t\}$ denote the risk set at time t, i.e., the set of individuals who are event-free and uncensored just prior to time t. To handle tied event times, we adopt the Breslow approximation [3]. Let $D(t) = \{i : t_i = t,\ \delta_i = 1\}$ denote the set of subjects experiencing the event at time t, and let $d(t) = |D(t)|$. The Breslow partial likelihood is given by

$$L^B(\beta) = \prod_t \frac{\prod_{i \in D(t)} \exp(\beta^\top \mathbf{X}_i(t))}{\left[\sum_{j \in R(t)} \exp(\beta^\top \mathbf{X}_j(t))\right]^{d(t)}}, \tag{2}$$

with log-partial likelihood $\ell^B(\beta) = \log L^B(\beta)$.

When the dimensionality increases, regularization is essential to obtain stable estimates and perform variable selection. Here, we consider the LASSO-penalized Cox model, which imposes an ℓ_1 penalty on the coefficients. The penalized objective function has form

$$Q(\beta) = \ell^B(\beta) - \lambda \sum_{j=1}^{p} |\beta_j|, \tag{3}$$

where $\lambda \ge 0$ controls the degree of shrinkage. The corresponding estimator is

$$\widehat{\beta}(\lambda) = \arg\max_{\beta \in \mathbb{R}^p} \left\{ \ell^B(\beta) - \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{4}$$

The ℓ_1 penalty encourages sparsity by shrinking some coefficients exactly to zero, providing automatic variable selection and yielding more interpretable survival models. Theoretical properties of the LASSO estimator for time-dependent Cox models, including oracle properties and asymptotic behavior, have been established by Bradic et al. [2].

3 Optimization Algorithm for the LASSO-Penalized Cox Model

The LASSO-penalized estimator defined in Section 2 maximizes a logpartial likelihood with an ℓ_1 penalty. While the objective function is concave, efficient optimization becomes challenging due to the non-differentiability of the penalty and the increasing computational burden as the number of covariates grows.

To address these challenges, the AACD algorithm provides an efficient framework for computing the penalized time-dependent Cox estimator. The method integrates coordinate descent updates with warm starts along the regularization path and an adaptive active-set strategy. This design enables the algorithm to concentrate computations on the most relevant variables, thereby improving computational efficiency as the number of covariates increases, without compromising estimation accuracy.

The estimator is obtained by maximizing the penalized logpartial likelihood in (3) using a cyclic coordinate descent scheme, where one coefficient is updated at a time while the others are held fixed. Let $\beta^{(k)}$ denote the current iterate. The update for the j-th coefficient is derived from a quadratic approximation of the logpartial likelihood centered at $\beta^{(k)}$:

$$\ell^B(\beta) \approx \ell^B(\beta^{(k)}) + U_j(\beta^{(k)})(\beta_j - \beta_j^{(k)}) - \frac{1}{2} I_{jj}(\beta^{(k)})(\beta_j - \beta_j^{(k)})^2, \tag{1}$$

where $\mathbf{U}(\beta) = \nabla \ell^B(\beta)$ is the score vector, and $\mathbf{I}(\beta) = -\nabla^2 \ell^B(\beta)$ is the observed information matrix (negative Hessian). The quantities $U_j(\beta)$ and $I_{jj}(\beta)$ denote the j-th component of the score vector and the j-th diagonal element of the information matrix, respectively.

Under the Breslow approximation, the score vector can be expressed as

$$\mathbf{U}(\beta) = \frac{\partial \ell^B(\beta)}{\partial \beta} = \sum_t \left[\sum_{i \in D(t)} \mathbf{X}_i(t) - d(t) \bar{\mathbf{X}}(t;\beta) \right], \tag{2}$$

where

$$\bar{\mathbf{X}}(t;\beta) = \frac{\sum_{j \in R(t)} \mathbf{X}_j(t) \exp\left(\beta^\top \mathbf{X}_j(t)\right)}{\sum_{j \in R(t)} \exp\left(\beta^\top \mathbf{X}_j(t)\right)}, \tag{3}$$

is the risk-set weighted average of the covariates at time t. The observed information matrix is given by

$$\mathbf{I}(\beta) = \sum_t d(t)\, \mathbf{V}(t;\beta), \tag{4}$$

where

$$\mathbf{V}(t;\beta) = \frac{\sum_{j \in R(t)} \mathbf{X}_j(t)\mathbf{X}_j(t)^\top \exp\left(\beta^\top \mathbf{X}_j(t)\right)}{\sum_{j \in R(t)} \exp\left(\beta^\top \mathbf{X}_j(t)\right)} - \bar{\mathbf{X}}(t;\beta)\bar{\mathbf{X}}(t;\beta)^\top. \tag{5}$$

Maximizing the quadratic approximation together with the ℓ_1 penalty leads to the coordinate-wise update

$$\beta_j^{(k+1)} = \mathscr{S}\left(\beta_j^{(k)} + \frac{U_j(\beta^{(k)})}{I_{jj}(\beta^{(k)})}, \frac{\lambda}{I_{jj}(\beta^{(k)})} \right), \tag{6}$$

where $\mathscr{S}(z,\gamma) = \text{sign}(z)(|z| - \gamma)_+$ is the soft-thresholding operator. This update can be interpreted as a proximal Newton step based on a diagonal approximation of the Hessian, where the quadratic expansion of the smooth logpartial likelihood is combined with the proximal mapping of the ℓ_1 penalty.

Estimation is performed along a decreasing sequence of regularization parameters $\Lambda = \{\lambda_1 > \cdots > \lambda_L\}$. Warm starts are used so that the solution for $\lambda_{\ell-1}$ initializes the algorithm for λ_ℓ. Convergence is declared when the maximum relative change in coefficients is below a tolerance ε_{CD}.

3.1 An Adaptive Active-Set Strategy

To reduce unnecessary computations, updates are restricted to an adaptively determined active set:

$$\mathscr{A}^{(k)} = \{ j : |\beta_j^{(k)}| > \varepsilon_{\text{active}} \}. \tag{7}$$

Algorithm 1: AACD algorithm for LASSO-penalized time-dependent Cox model

Input: Counting-process data $(\texttt{start}, \texttt{stop}, \delta)$ and covariates $\mathbf{X}$;
Regularization path $\lambda_1 > \cdots > \lambda_L$;
Maximum coordinate descent iterations K_{CD} and tolerance $\varepsilon_{\mathrm{CD}}$
Output: Regularization path $\{\widehat{\beta}(\lambda_\ell)\}_{\ell=1}^{L}$ and selected estimator $\widehat{\beta}(\widehat{\lambda})$

Initialize $\widehat{\beta}(\lambda_1) = \mathbf{0}$;
for $\ell = 2, \ldots, L$ **do**
  Set $\beta^{(0)} = \widehat{\beta}(\lambda_{\ell-1})$;
  Define active set $\mathscr{A} = \{j : |\beta_j^{(0)}| > \varepsilon_{\mathrm{active}}\}$;
  for $m = 1, \ldots, K_{\mathrm{CD}}$ **do**
    Compute score vector $\mathbf{U}$ and diagonal information matrix $\mathbf{I}$;
    foreach $j \in \mathscr{A}$ **do**
      $\beta_j \leftarrow \mathscr{S}\left(\beta_j + \frac{U_j}{I_{jj}}, \frac{\lambda_\ell}{I_{jj}}\right)$;
    foreach $j \notin \mathscr{A}$ **do**
      if $|U_j| > \lambda_\ell$ **then**
        add j to $\mathscr{A}$;
    Update active set $\mathscr{A} = \{j : |\beta_j| > \varepsilon_{\mathrm{active}}\}$;
    if *maximum relative change* $< \varepsilon_{\mathrm{CD}}$ **then**
      break
  $\widehat{\beta}(\lambda_\ell) = \beta$;
Compute $\mathrm{EBIC}(\lambda_\ell)$ for $\ell = 1, \ldots, L$ and set $\widehat{\lambda} = \arg\min_\ell \mathrm{EBIC}(\lambda_\ell)$;
return $\{\widehat{\beta}(\lambda_\ell)\}_{\ell=1}^{L}$ and $\widehat{\beta}(\widehat{\lambda})$;

where ε is a small threshold parameter used to define the adaptive active set. Based on the sensitivity analysis reported in Section 4.4, we used $\varepsilon_{\mathrm{active}} = 10^{-10}$ in all simulation experiments.

During each sweep, only coefficients in $\mathscr{A}^{(k)}$ are updated. Inactive variables are screened using a KKT-type condition based on their score contributions; specifically, variables with $|U_j| > \lambda$ are added to the active set. This mechanism concentrates computation on variables most likely to have nonzero effects, improving efficiency without compromising estimation accuracy.

Model adequacy is evaluated using the Breslow logpartial likelihood (2). For model selection, we employ the extended Bayesian information criterion (EBIC) given by

$$\mathrm{EBIC}(\lambda) = -2\,\ell^{B}(\widehat{\beta}(\lambda)) + \log(d)\,df(\lambda) + 2\gamma\log(p)\,df(\lambda), \tag{8}$$

where $d = \sum_i \delta_i$, $\gamma \in [0,1]$, and

$$df(\lambda) = \sum_{j=1}^{p} \frac{|\widehat{\beta}_j(\lambda)|}{|\widehat{\beta}_j(\lambda)| + \varepsilon_{\mathrm{active}}}. \tag{9}$$

which provides a smooth approximation to the number of nonzero coefficients. The optimal λ is selected as $\widehat{\lambda} = \arg\min_{\lambda\in\Lambda} \mathrm{EBIC}(\lambda)$. Algorithm 1 summarizes our proposed estimation procedure.

3.2 Convergence Analysis of the AACD Algorithm

In this section, we discuss the convergence properties of the proposed AACD algorithm. Let

$$Q(\beta) = \ell^{B}(\beta) - \lambda \sum_{j=1}^{p} |\beta_j|$$

denote the LASSO-penalized Breslow partial log-likelihood, where $\ell^B(\beta)$ is concave and continuously differentiable on a compact parameter space, and the ℓ_1 penalty is convex and coordinate-wise separable.

The KKT optimality conditions imply that $\widehat{\beta}$ is a global maximizer of $Q(\beta)$ if and only if

$$U_j(\widehat{\beta}) = \lambda \operatorname{sign}(\widehat{\beta}_j), \quad \text{for } \widehat{\beta}_j \neq 0,$$

and

$$|U_j(\widehat{\beta})| \leq \lambda, \quad \text{for } \widehat{\beta}_j = 0,$$

where $U_j(\beta)$ denotes the jth component of the score vector.

Theorem 1. *Suppose that the negative Breslow partial log-likelihood $-\ell^B(\beta)$ is strictly convex and the adaptive active-set strategy ensures that every coordinate violating the global KKT conditions is reintroduced into the active set within a finite number of iterations. Then, under standard regularity conditions for proximal Newton-type coordinate descent methods, the sequence of iterates generated by the AACD algorithm converges to the unique global maximizer $\widehat{\beta}(\lambda)$ of the LASSO-penalized Cox objective function.*

Proof. The AACD algorithm combines cyclic coordinate descent with an adaptive active-set strategy. At each iteration, a quadratic local minorant of the smooth partial log-likelihood function is constructed for the active coordinates. The resulting one-dimensional subproblem is then maximized exactly using the soft-thresholding operator. Under the assumption of a bounded Hessian for $\ell^B(\beta)$, these coordinate-wise updates guarantee a monotonic non-decreasing improvement of the penalized objective function $Q(\beta)$ at each step.

The adaptive active-set mechanism improves computational efficiency by restricting updates to promising coordinates while preserving global convergence through systematic KKT screening. Any inactive variable violating the global KKT condition,

$$|U_j(\beta)| > \lambda,$$

is immediately reintroduced into the active set and subsequently updated in the next cycle. Therefore, no coordinate violating the optimality conditions is permanently excluded from the optimization process.

Since the objective function consists of a continuously differentiable concave component and a coordinate-wise separable non-smooth penalty, and because every violating coordinate is visited infinitely often through the cyclic KKT screening mechanism, the convergence framework established by Tseng [18] for coordinate-wise optimization directly applies. Consequently, the sequence of iterates generated by the AACD algorithm is guaranteed to converge to the unique global maximizer $\widehat{\beta}(\lambda)$. □

3.3 Computational Complexity

We analyze the computational complexity of the AACD algorithm. Let K_{CD} be the number of coordinate descent iterations required for convergence at a given λ. Let $\mathscr{A}^{(k)}$ denote the active set at iteration k, and $|\mathscr{A}|$ its typical size near convergence.

Theorem 2. *Under the Breslow approximation, the per-iteration complexity of the AACD algorithm is $\mathscr{O}(np)$ in the worst case and $\mathscr{O}(n|\mathscr{A}|+p)$ when updates are restricted to an active set with $|\mathscr{A}| \ll p$. Consequently, the total complexity is $\mathscr{O}(LK_{\mathrm{CD}}np)$ in the worst case and $\mathscr{O}(LK_{\mathrm{CD}}(n|\mathscr{A}|+p))$ in practice.*

Compared with standard coordinate descent methods that update all p coordinates at every iteration and therefore require $\mathscr{O}(np)$ operations per sweep, the proposed AACD algorithm restricts updates to an adaptive active set. Consequently, when the active set size satisfies $|\mathscr{A}| \ll p$, the practical computational cost is substantially reduced to $\mathscr{O}(n|\mathscr{A}|+p)$. This improvement becomes especially important in sparse time-dependent Cox models where only a relatively small subset of variables remains active along the regularization path.

Proof. We analyze the per-iteration cost of the AACD algorithm under the Breslow approximation by examining its main computational components. At each iteration, evaluating the score vector $\mathbf{U}(\beta^{(k)})$ requires computing, for each distinct event time t, the risk-set weighted average

$$\bar{\mathbf{X}}(t;\beta) = \frac{\sum_{j\in R(t)} \mathbf{X}_j(t) \exp\left(\beta^\top \mathbf{X}_j(t)\right)}{\sum_{j\in R(t)} \exp\left(\beta^\top \mathbf{X}_j(t)\right)}.$$

A full evaluation over all n subjects and p covariates costs $\mathcal{O}(np)$. However, since only the coordinates $\{\beta_j : j \in \mathscr{A}^{(k)}\}$ are updated at iteration k, the linear predictors $\eta_j(t) = \beta^\top \mathbf{X}_j(t)$ change only through these components. Consequently, the exponential weights $w_j(t) = \exp(\eta_j(t))$ for $j \notin \mathscr{A}^{(k)}$ remain unchanged, and the aggregates

$$S^{(0)}(t;\beta) = \sum_{j \in R(t)} w_j(t), \qquad S^{(1)}(t;\beta) = \sum_{j \in R(t)} \mathbf{X}_j(t)\, w_j(t)$$

can be updated incrementally by recomputing only the contributions associated with $\mathscr{A}^{(k)}$, for example via

$$S^{(0)}_{\text{new}}(t) = S^{(0)}_{\text{old}}(t) + \sum_{j \in \mathscr{A}^{(k)} \cap R(t)} \left[w_j^{\text{new}}(t) - w_j^{\text{old}}(t) \right],$$

which, when summed over at most n event times, leads to a cost of order $\mathcal{O}(n|\mathscr{A}^{(k)}|)$ when incremental updates are used, compared to $\mathcal{O}(np)$ for full recomputation. The coordinate descent step updates only $j \in \mathscr{A}^{(k)}$ via the soft-thresholding operation

$$\beta_j^{(k+1)} = \mathscr{S}\left(\beta_j^{(k)} + \frac{U_j(\beta^{(k)})}{I_{jj}(\beta^{(k)})}, \frac{\lambda}{I_{jj}(\beta^{(k)})} \right),$$

contributing $\mathcal{O}(|\mathscr{A}^{(k)}|)$ operations per sweep. After each sweep, inactive variables $j \notin \mathscr{A}^{(k)}$ are screened via the condition $|U_j(\beta^{(k)})| > \lambda$; since each U_j can be evaluated using the maintained aggregates $S^{(0)}$ and $S^{(1)}$ with constant cost per coordinate, this step incurs $\mathcal{O}(p)$ operations in total. Combining these components yields a per-iteration cost of $\mathcal{O}(np)$ in the worst case, when full score recomputation is required and $|\mathscr{A}^{(k)}| = p$, and $\mathcal{O}(n|\mathscr{A}^{(k)}| + p)$ when incremental updates are used and $|\mathscr{A}^{(k)}| \ll p$. Multiplying by the number of coordinate descent iterations K_{CD} required for convergence and extending over L values of λ along the regularization path yields the stated total complexity $\mathcal{O}(LK_{\text{CD}}np)$ in the worst case and $\mathcal{O}(LK_{\text{CD}}(n|\mathscr{A}| + p))$ in practice, where $|\mathscr{A}|$ denotes the typical active-set size near convergence. □

4 Simulation Study

We conduct a Monte Carlo simulation study to evaluate the finite-sample performance of the proposed AACD algorithm for LASSO-penalized estimation in time-dependent Cox proportional hazards models. The study aims to assess both statistical accuracy and computational efficiency relative to the widely used `glmnet` implementation.

All datasets are generated in the $(\text{start}, \text{stop}, \text{event})$ counting-process format to ensure that risk sets are correctly constructed in the presence of time-dependent covariates. Survival times follow the time-dependent Cox model

$$h(t \mid \mathbf{X}(t)) = h_0(t) \exp\{\beta_0^\top \mathbf{X}(t)\}, \qquad h_0(t) \equiv 1,$$

where $\beta_0 = (\beta_{01}^\top, \beta_{02}^\top)^\top$ is the true coefficient vector. The model is sparse with $p_1 = 6$ nonzero coefficients $\beta_{01} = (1.2, 0.8, 0.5, -0.5, -0.8, -1.2)^\top$ and $p_2 = p - p_1$ zero coefficients $\beta_{02} = \mathbf{0}_{p_2}$.

We consider sample sizes $n \in \{200, 400, 600\}$ and dimensionality levels $p \in \{10, 20, 30\}$. Covariates are drawn from a multivariate normal distribution with zero mean and AR(1) correlation structure $\Sigma_{ij} = \rho^{|i-j|}$, with $\rho \in \{0, 0.25, 0.5\}$. Time dependence is induced by partitioning each subject's follow-up into a fixed number of intervals and updating covariates at interval boundaries. Event times are generated using the inverse probability transform, and independent censoring times are sampled from an exponential distribution to achieve approximate censoring proportions $p_c \in \{0, 0.2, 0.4\}$.

For each dataset, the LASSO-penalized Cox estimator is computed along a decreasing regularization path using the AACD coordinate descent algorithm with warm starts. The tuning parameter is selected by minimizing an EBIC-type criterion (Equation 8). For comparison, the LASSO estimator is also computed using `glmnet`, where the regularization parameter is selected via 3-fold cross-validation.

The choice of these tuning mechanisms is motivated by the computational characteristics of time-dependent survival data. While `glmnet` conventionally relies on K-fold cross-validation, this approach necessitates repeated risk-set reconstructions across multiple folds, which becomes computationally burdensome as the number of predictors and observations increases. In contrast, the AACD algorithm utilizes an EBIC-type criterion to bypass this bottleneck, providing a more efficient and scalable solution for time-dependent survival analysis.

Each simulation configuration is replicated $m = 50$ times, covering all combinations of sample size n, dimensionality p, correlation level ρ, and censoring proportion p_c. Performance is evaluated in terms of variable selection, estimation accuracy, and computational efficiency. Let $\mathscr{S}_0 = \{j : \beta_{0j} \neq 0\}$ denote the true active set with $|\mathscr{S}_0| = p_1$, and let $\widehat{\mathscr{S}} = \{j : \widehat{\beta}_j \neq 0\}$ denote the estimated active set. Variable selection errors are quantified using false positives (FP) and false negatives (FN), defined respectively as $\mathrm{FP} = |\widehat{\mathscr{S}} \setminus \mathscr{S}_0|, \quad \mathrm{FN} = |\mathscr{S}_0 \setminus \widehat{\mathscr{S}}|$.

Estimation accuracy is assessed via the mean squared error (MSE) of the regression coefficients. To compare AACD directly with `glmnet`, we define the simulated relative efficiency (SREF) as

$$\mathrm{SREF} = \frac{\mathrm{MSE}(\widehat{\beta}_{\texttt{glmnet}})}{\mathrm{MSE}(\widehat{\beta}_{\mathrm{AACD}})},$$

where values larger than one indicate superior performance of AACD. Computational efficiency is evaluated using wall-clock runtime (in seconds) measured under identical hardware and software conditions.

4.1 Estimation Accuracy

Table 1 presents the SREF of the AACD estimator with respect to `glmnet`, computed coefficient-wise as the ratio of the empirical mean squared errors (MSEs) of the estimated active coefficients. Values larger than one indicate that AACD achieves smaller MSEs and therefore provides more accurate estimation. The analysis focuses on the six nonzero coefficients in the active set, $\beta_{01} = (\beta_1, \ldots, \beta_6)$, with active dimensionality $p_1 = 6$.

Table 1. The SREF of AACD relative to `glmnet`, based on the ratio of mean squared errors (MSEs) for the active coefficients.

	$n = 200$			$n = 400$			$n = 600$		
β	$p = 10$	$p = 20$	$p = 30$	$p = 10$	$p = 20$	$p = 30$	$p = 10$	$p = 20$	$p = 30$
β_1	1.59	1.21	1.16	1.34	1.06	1.14	1.25	1.06	1.04
β_2	1.55	1.29	1.17	1.47	1.10	1.17	1.41	1.13	1.09
β_3	1.37	1.17	1.10	1.26	1.07	1.12	1.18	1.03	1.04
β_4	1.35	1.17	1.12	1.24	1.07	1.13	1.13	1.05	1.03
β_5	1.57	1.23	1.19	1.40	1.16	1.17	1.38	1.19	1.12
β_6	1.42	1.21	1.15	1.26	1.06	1.08	1.15	1.03	1.01

Overall, AACD consistently outperforms `glmnet` across all simulation settings. The relative efficiency values exceed one for all coefficients and combinations of sample size n and dimensionality p, indicating consistently smaller empirical MSEs.

The efficiency gains are most pronounced in smaller samples. For $n = 200$, SREF ranges approximately from 1.10 to 1.59, implying an MSE reduction of up to approximately 37% relative to `glmnet`. As the sample size increases, the advantage diminishes but remains substantial: for $n = 400$, SREF mostly lies between 1.06 and 1.47, while for $n = 600$ it ranges roughly from 1.01 to 1.41.

A similar trend is observed as dimensionality increases. When $p = 10$, AACD achieves the largest improvement. As p grows to 20 and 30, the efficiency gains decrease but remain consistently above one, indicating robust performance even in higher-dimensional settings.

Table 2 provides additional insight into the estimation behavior of the two methods by reporting the empirical bias and simulation standard error (SE) of the estimated active coefficients over the Monte Carlo replications. Across nearly all settings, AACD exhibits smaller absolute bias than `glmnet` for each of the six active coefficients. This difference is particularly pronounced for smaller sample sizes, where the shrinkage effect of `glmnet` leads to noticeably larger bias toward zero. As the sample size increases from $n = 200$ to $n = 600$, the magnitude of bias decreases for both methods, indicating improved estimation consistency in larger samples.

The reported simulation SEs are generally small for both procedures and tend to decrease as the sample size increases, reflecting the expected gain in stability with larger datasets. Importantly, AACD achieves lower bias while maintaining variability comparable to that of `glmnet`. This suggests that the improvement in estimation accuracy is not obtained at the expense of increased instability. The results therefore support the conclusion that AACD provides more reliable coefficient estimation, particularly in finite-sample settings where bias reduction is critical.

Table 2. Biases (SEs) of the estimated active coefficients for AACD and GLMNET under different sample sizes (n) and dimensions (p).

Method	n	p	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$
AACD	200	10	−0.288 (0.027)	−0.203 (0.024)	−0.217 (0.021)	0.227 (0.020)	0.197 (0.024)	0.302 (0.025)
GLMNET	200	10	−0.438 (0.026)	−0.322 (0.021)	−0.316 (0.019)	0.321 (0.018)	0.312 (0.022)	0.435 (0.025)
AACD	200	20	−0.430 (0.024)	−0.301 (0.022)	−0.309 (0.017)	0.310 (0.017)	0.312 (0.024)	0.426 (0.024)
GLMNET	200	20	−0.500 (0.024)	−0.357 (0.023)	−0.355 (0.016)	0.357 (0.016)	0.363 (0.024)	0.493 (0.025)
AACD	200	30	−0.484 (0.023)	−0.338 (0.022)	−0.346 (0.015)	0.334 (0.016)	0.347 (0.021)	0.463 (0.023)
GLMNET	200	30	−0.534 (0.024)	−0.377 (0.021)	−0.375 (0.015)	0.369 (0.015)	0.386 (0.022)	0.509 (0.025)
AACD	400	10	−0.265 (0.019)	−0.176 (0.017)	−0.198 (0.015)	0.199 (0.015)	0.182 (0.016)	0.275 (0.018)
GLMNET	400	10	−0.373 (0.019)	−0.262 (0.015)	−0.275 (0.015)	0.274 (0.015)	0.264 (0.015)	0.373 (0.018)
AACD	400	20	−0.341 (0.019)	−0.240 (0.017)	−0.259 (0.013)	0.260 (0.013)	0.237 (0.018)	0.344 (0.019)
GLMNET	400	20	−0.390 (0.018)	−0.275 (0.016)	−0.295 (0.014)	0.295 (0.014)	0.279 (0.016)	0.389 (0.017)
AACD	400	30	−0.371 (0.018)	−0.258 (0.016)	−0.283 (0.013)	0.278 (0.012)	0.258 (0.016)	0.390 (0.017)
GLMNET	400	30	−0.416 (0.018)	−0.297 (0.015)	−0.318 (0.013)	0.316 (0.013)	0.293 (0.015)	0.428 (0.018)
AACD	600	10	−0.237 (0.016)	−0.160 (0.015)	−0.186 (0.013)	0.186 (0.013)	0.161 (0.014)	0.256 (0.017)
GLMNET	600	10	−0.327 (0.016)	−0.234 (0.014)	−0.247 (0.013)	0.246 (0.013)	0.230 (0.012)	0.334 (0.017)
AACD	600	20	−0.307 (0.017)	−0.200 (0.015)	−0.225 (0.012)	0.222 (0.012)	0.205 (0.015)	0.300 (0.016)
GLMNET	600	20	−0.351 (0.016)	−0.240 (0.013)	−0.260 (0.013)	0.260 (0.013)	0.249 (0.013)	0.344 (0.015)
AACD	600	30	−0.331 (0.015)	−0.227 (0.014)	−0.249 (0.011)	0.247 (0.012)	0.225 (0.014)	0.334 (0.015)
GLMNET	600	30	−0.366 (0.015)	−0.257 (0.012)	−0.276 (0.012)	0.276 (0.012)	0.259 (0.012)	0.365 (0.015)

Finally, the improvement is stable across all six coefficients, suggesting that the efficiency gains reflect a general enhancement in estimation accuracy rather than the effect of specific parameters. In summary, AACD provides systematically more efficient coefficient estimates than `glmnet`, particularly in small-sample and low-dimensional regimes, while remaining competitive as both sample size and dimensionality increase.

4.2 Variable Selection Performance

Figures 1 and 2 display heatmaps of mean FP and FN values for the AACD and `glmnet` across different simulation settings, varying sample size n, censoring proportion p_c, and correlation level ρ.

Figure 1 shows that AACD generally produces slightly higher FP values than `glmnet`, indicating selection of more irrelevant variables. This tendency is most pronounced under low correlation among predictors ($\rho = 0$ or 0.25), where AACD includes more noise variables. As correlation increases ($\rho = 0.5$), the difference diminishes, and AACD occasionally produces FP values comparable to or lower than `glmnet`. The FP pattern is relatively stable across sample sizes and censoring proportions.

Figure 2 demonstrates that the AACD typically achieves lower FN values than `glmnet`, particularly under moderate or high censoring. In low to moderate correlation settings ($\rho = 0$ or 0.25), AACD accurately identifies almost all truly active variables, yielding FN values close to zero. With stronger correlation ($\rho = 0.5$), FN values increase for both methods, reflecting the challenge of variable selection in highly correlated scenarios. Nonetheless, AACD tends to retain more true signals than `glmnet`, especially for smaller sample sizes.

Overall, these results reveal a trade-off between the methods: AACD favors recovering the true active set, resulting in fewer false negatives, while `glmnet` is more conservative, yielding fewer false positives. From a practical standpoint, AACD's approach is advantageous in applications where failing to select important predictors is more costly than including a few extra variables.

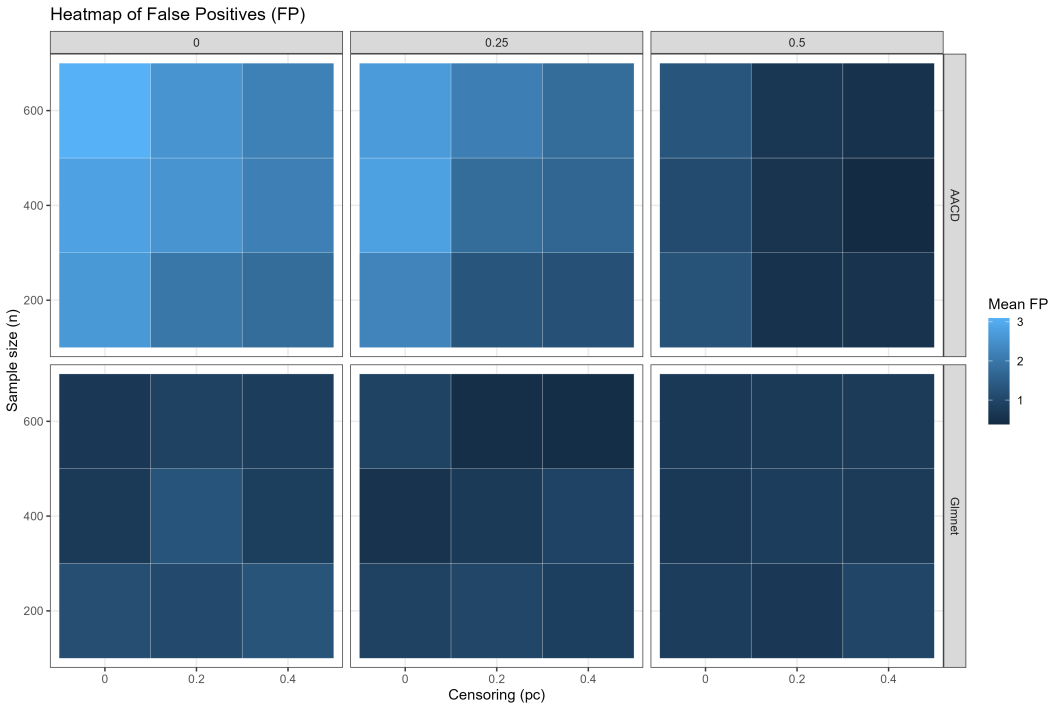

Figure 1. Heatmap of mean false positives (FP) for AACD and `glmnet` across sample size n, censoring proportion p_c, and correlation ρ.

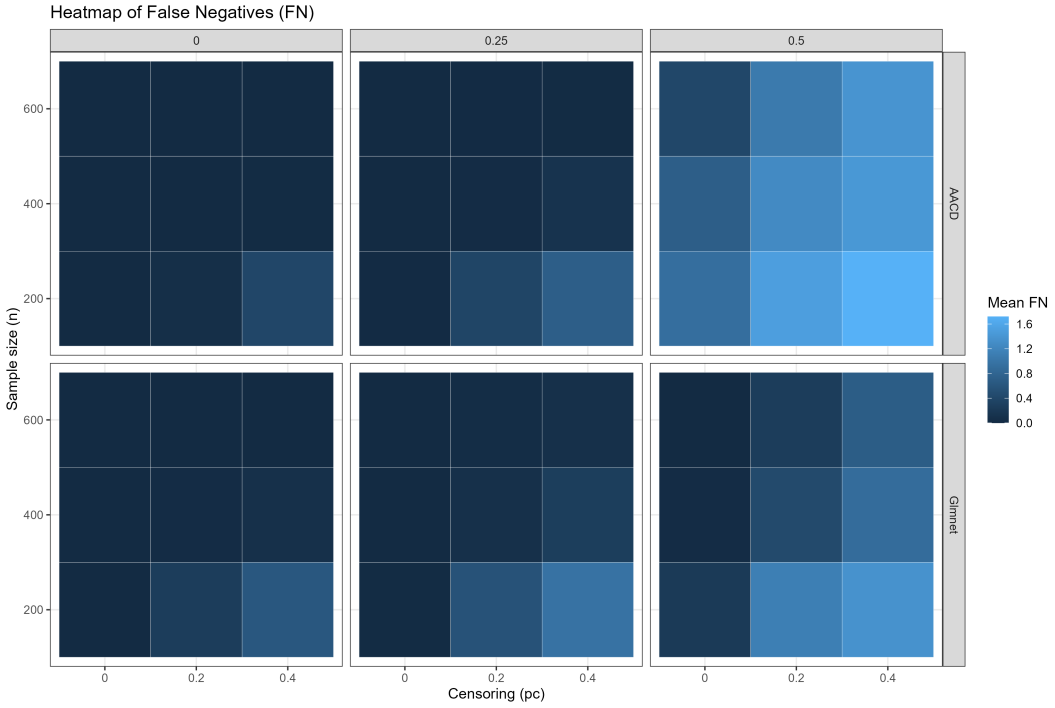

Figure 2. Heatmap of mean false negatives (FN) for AACD and `glmnet` across sample size n, censoring proportion p_c, and correlation ρ.

4.3 Computational Efficiency

Figures 3 and 4 summarize the computational performance of AACD and `glmnet` across different sample sizes n, censoring proportions p_c, and correlation levels ρ. Figure 3 presents the mean runtime, while Figure 4 reports the relative efficiency, defined as

$$\mathrm{RE} = \frac{\mathrm{Runtime}_{\texttt{glmnet}}}{\mathrm{Runtime}_{\mathrm{AACD}}},$$

where values greater than one indicate that AACD is faster than `glmnet`.

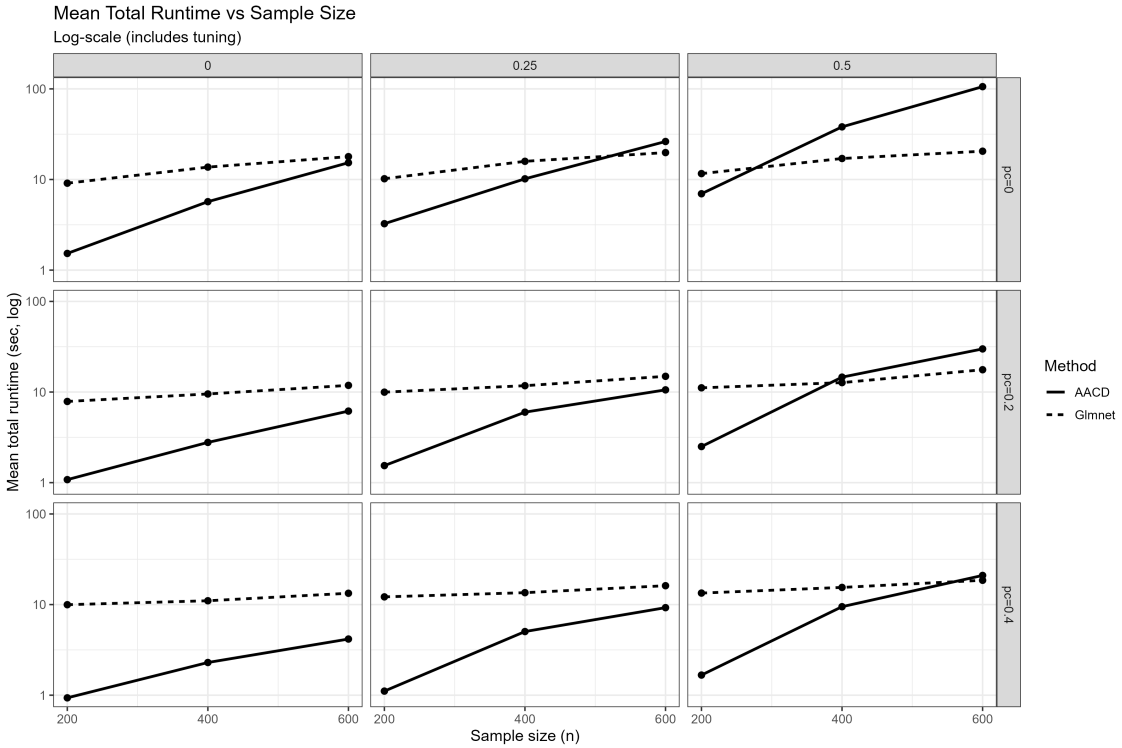

Figure 3. Mean total runtime of AACD and `glmnet` versus sample size n, faceted by censoring proportion (p_c) and correlation (ρ).

When covariates are independent ($\rho = 0$), AACD consistently requires less computational time than `glmnet` across all sample sizes and censoring levels. Although runtime increases with sample size for both methods, AACD remains substantially faster. The relative efficiency is consistently above one and tends to increase with the censoring proportion, highlighting AACD's computational advantage in low-correlation settings.

For moderate correlation ($\rho = 0.25$), runtimes increase for both methods, but AACD still outperforms `glmnet` in most scenarios. When censoring is present ($p_c = 0.2$ or 0.4), AACD maintains shorter runtimes across all sample sizes, and the relative efficiency remains above one in most cases. Only in the absence of censoring and at large sample sizes does the performance gap narrow, with `glmnet` occasionally performing slightly better.

Under strong correlation ($\rho = 0.5$), AACD's runtime increases rapidly without censoring ($p_c = 0$), eventually exceeding that of `glmnet`. This pattern is reflected in the relative efficiency falling below one. However, introducing censoring ($p_c = 0.2$ or 0.4) partially restores AACD's advantage, particularly for smaller sample sizes, with relative efficiency approaching or exceeding one.

Table 3 provides an additional comparison of computational runtime under different tuning strategies. To separate the effect of the optimization algorithm from that of the tuning criterion, `glmnet` was evaluated using both EBIC and cross-validation. The results show that the use of EBIC substantially reduces runtime for both methods compared with cross-validation, confirming that tuning strategy plays an important role in computational cost. However, AACD combined with EBIC generally achieves lower runtime than `glmnet` with the same EBIC-based tuning, particularly under weak and moderate correlation structures ($\rho = 0$ and $\rho = 0.25$). Under strong correlation ($\rho = 0.5$), the computational gap becomes smaller and `glmnet` with EBIC is slightly faster on average. Nevertheless, AACD maintains

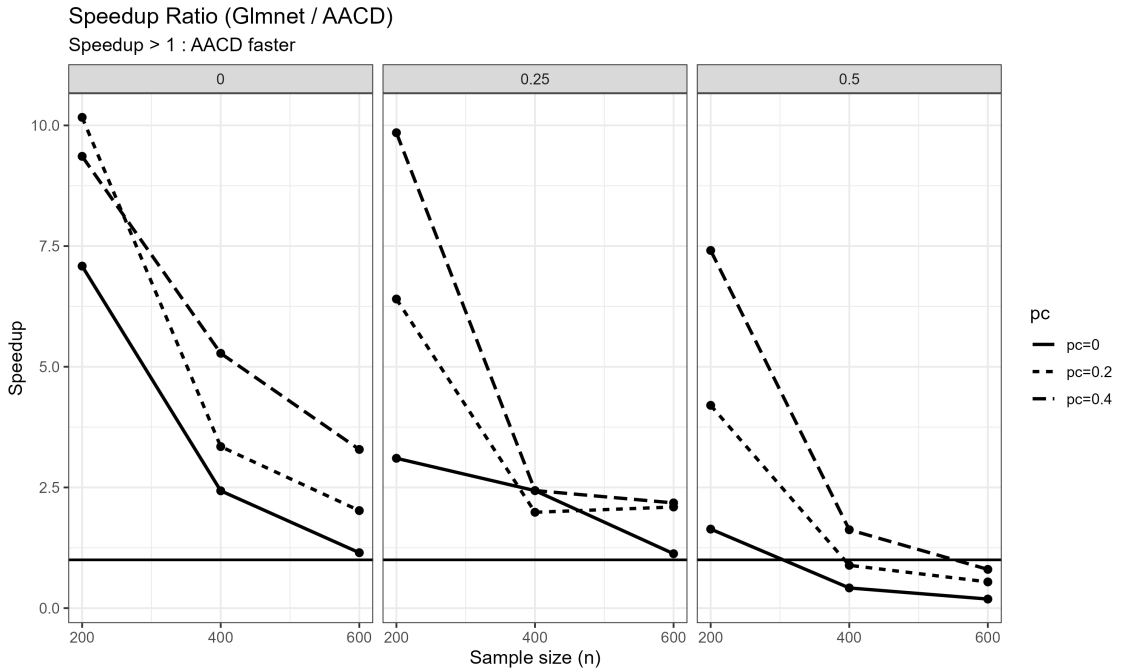

Figure 4. Relative efficiency (glmnet / AACD) of mean total runtime versus sample size n, faceted by correlation ρ; values above one indicate AACD is faster.

competitive performance overall while avoiding the large computational burden associated with repeated cross-validation. These findings indicate that the computational advantage of AACD is attributable not only to the EBIC tuning strategy but also to the efficiency of the proposed optimization framework.

Table 3. Average runtime comparison under different tuning strategies

Method	$\rho = 0$	$\rho = 0.25$	$\rho = 0.50$	Overall
AACD (EBIC)	0.73 ± 0.18	1.39 ± 0.66	3.04 ± 0.68	1.72 ± 0.51
glmnet (EBIC)	1.94 ± 0.28	2.31 ± 0.35	2.59 ± 0.66	2.28 ± 0.43
glmnet (CV)	6.99 ± 0.85	10.52 ± 0.82	12.76 ± 0.73	10.08 ± 0.80

4.4 Sensitivity Analysis for Active-Set Threshold

To investigate the influence of the active-set threshold parameter $\varepsilon_{\text{active}}$, we conducted an additional sensitivity analysis under a representative simulation setting ($n = 200$, $p = 10$, $\rho = 0.25$, $p_c = 0.2$). The parameter $\varepsilon_{\text{active}}$ controls whether a coefficient is retained in the active coordinate set during optimization. Table 4 summarizes the resulting mean FP, mean FN, and average runtime over 50 replications for different threshold values.

The results indicate that the AACD algorithm is relatively robust to the choice of $\varepsilon_{\text{active}}$. Extremely large threshold values ($\varepsilon_{\text{active}} \geq 0.1$) lead to slightly larger FP values, suggesting that overly aggressive screening may reduce selection stability. In contrast, moderate and small thresholds (10^{-2} to 10^{-12}) produce highly similar runtimes and variable-selection performance. The best overall balance was observed around $\varepsilon_{\text{active}} = 10^{-10}$, which achieved the lowest FP and FN values without increasing computational cost substantially. These findings suggest that the proposed active-set strategy improves computational scalability while maintaining stable statistical performance across a broad range of threshold choices.

Table 4. Sensitivity analysis of the AACD active-set threshold ε

$\varepsilon_{\text{active}}$	Mean FP	Mean FN	Mean Runtime (s)
1	3.62	0.00	0.761
10^{-2}	3.30	0.06	0.821
10^{-4}	2.54	0.02	0.806
10^{-6}	2.84	0.08	0.935
10^{-8}	2.88	0.04	0.933
10^{-10}	2.44	0.00	0.759
10^{-12}	2.50	0.02	0.799
10^{-14}	2.96	0.04	0.851

5 Mayo Clinic PBC Data Analysis

To evaluate the practical performance of the proposed method, we analyze the Mayo Clinic primary biliary cirrhosis (PBC) dataset [14], a widely used benchmark in survival analysis. After excluding incomplete observations, the final sample includes 312 patients with longitudinal follow-up. The data were transformed into counting-process form with $(t_{\text{start}}, t_{\text{stop}}]$ intervals using the `tmerge()` function to accommodate time-dependent covariates [15]. The outcome is time to death, with transplantation and censoring treated accordingly. The covariates consist of demographic variables and repeatedly measured clinical biomarkers; a detailed description is provided in Table 5. All continuous covariates were transformed and standardized prior to model fitting.

Table 5. Variables in the PBC time-dependent dataset

Variable	**Description**
id	Patient identifier
day	Follow-up time (days)
time	Time to event/censoring (days)
status	Event indicator (0=censored, 1=transplant, 2=death)
bili	Serum bilirubin
albumin	Serum albumin
protime	Prothrombin time
alk.phos	Alkaline phosphatase
ast	Aspartate aminotransferase
chol	Serum cholesterol
platelet	Platelet count
ascites	Ascites indicator
edema	Edema score
hepato	Hepatomegaly indicator
spiders	Spider angiomas indicator
stage	Histologic stage
sex	Gender

Table 6 summarizes the estimated coefficients, their standard errors, and the relative efficiencies of the proposed AACD method compared to LASSO. Both methods identify similar key predictors, including *log(bili)*, *log(protime)*, *albumin*, and *stage*. However, AACD produces a more parsimonious model by shrinking several coefficients exactly to zero, whereas LASSO retains additional variables with small effects.

From an applied perspective, time-dependent variable selection enhances the interpretability of the fitted Cox model. Unlike standard Cox models with static covariates, the selected variables in this framework reflect the evolving relevance of clinical predictors over the follow-up period. This allows researchers to distinguish between covariates with persistent effects on the hazard function and those whose influence is transient. In the context of the PBC study, this yields a more clinically meaningful representation of disease progression, as only

the most relevant biomarkers are retained in the final sparse model.

Table 6. Estimated coefficients, standard errors, and relative efficiency (LASSO/AACD) for the time-dependent Cox model on the PBC dataset

	AACD		LASSO (glmnet)		
Variable	Coef	SE	Coef	SE	Relative Efficiency
log(bili)	1.1514	0.1528	1.2298	0.1655	1.0834
log(protime)	0.2327	0.0795	0.2287	0.0825	1.0382
log(alk.phos + 1)	0	–	-0.0171	0.1324	–
log(ast + 1)	-0.1658	0.0936	-0.1968	0.0945	1.0092
albumin	-0.7588	0.1142	-0.7830	0.1157	1.0126
chol	0	–	0	–	–
platelet	-0.0538	0.1147	-0.0623	0.1182	1.0305
stage	0.1329	0.1286	0.1677	0.1300	1.0108
ascites	0.0740	0.0700	0.0728	0.0710	1.0134
edema	0.1791	0.0932	0.1794	0.0963	1.0326
hepato	0	–	-0.0576	0.1265	–
spiders	0	–	0	–	–
sexf	-0.1051	0.0876	-0.1268	0.0914	1.0441

In terms of estimation accuracy, AACD consistently yields smaller standard errors for most selected variables, resulting in relative efficiency values greater than one. This indicates that the proposed method provides more stable and precise estimates compared to LASSO.

The predictive performance and computational efficiency of the two approaches are summarized in Table 7. Predictive accuracy was evaluated using the concordance index (C-index) [9], which measures the ability of the model to correctly rank survival times, together with the time-dependent area under the ROC curve (AUC) [10] at 1-, 3-, and 5-year horizons. To assess the stability of the results, bootstrap replications were performed and both the mean and standard deviation (SD) of each metric are reported.

Table 7. Comparison of predictive performance and computational efficiency between AACD and LASSO on the PBC dataset (mean $\pm$ SD over bootstrap replications)

Method	Runtime (s)	C-index	AUC (1-year)	AUC (3-year)	AUC (5-year)
AACD	16.57 ± 1.62	0.8719 ± 0.0179	0.8903 ± 0.0428	0.8938 ± 0.0206	0.8473 ± 0.0223
LASSO	45.72 ± 7.70	0.8745 ± 0.0180	0.8916 ± 0.0443	0.8962 ± 0.0205	0.8505 ± 0.0215

The results indicate that AACD and LASSO achieve highly comparable predictive discrimination across all evaluation criteria. In particular, the C-index and time-dependent AUC values are nearly identical at each prediction horizon, while the associated standard deviations remain small, demonstrating stable predictive performance across bootstrap samples. These findings suggest that the enhanced sparsity and estimation accuracy obtained by AACD do not come at the expense of predictive capability.

A major advantage of AACD is its computational efficiency. AACD requires substantially less computation time than LASSO, with an average runtime reduction of approximately 64%. Moreover, AACD also exhibits smaller runtime variability across bootstrap replications, indicating more stable computational behavior.

Overall, the proposed AACD method achieves an effective balance between sparsity, estimation accuracy, predictive performance, and computational efficiency. The bootstrap-based results further confirm the robustness and stability of the proposed approach for time-dependent Cox regression with longitudinal covariates.

6 Limitations of AACD

Despite its favorable computational and variable selection performance, the proposed AACD algorithm has several limitations. First, when covariates exhibit very strong correlation structures, the active set may grow substantially during optimization, reducing the computational advantage of AACD over standard coordinate descent methods. This behavior was also observed in the simulation study under $\rho = 0.5$, where the runtime advantage became less pronounced.

Second, the proposed framework is primarily designed for sparse or moderately sparse models. In dense signal settings, where many covariates have nonzero effects, the active-set mechanism becomes less effective because a larger proportion of variables must be updated at each iteration. Consequently, both computational efficiency and sparsity gains may diminish.

Finally, in ultra-high-dimensional settings with $p \gg n$, the stability of variable selection may become more sensitive to the tuning parameter and correlation structure, as commonly observed in penalized Cox regression methods. Although AACD remains computationally feasible in such settings, additional screening or dimensionality reduction strategies may further improve scalability and stability.

7 Conclusion

We introduce the AACD algorithm for LASSO-penalized time-dependent Cox models, combining coordinate descent, warm starts, and an adaptive active-set strategy. Simulation studies show that AACD achieves up to a 37% lower mean squared error than `glmnet` for small samples ($n = 200$) and consistently reduces false negatives across all settings, while maintaining competitive false positive rates. Computationally, AACD reduces runtime by up to 36% compared to the standard LASSO implementation in `glmnet`.

Application to the Mayo Clinic PBC dataset demonstrates that AACD produces more parsimonious models with smaller standard errors, while predictive performance remains comparable (C-index ≈ 0.87, 1-year AUC ≈ 0.88). These results highlight AACD as a practical and efficient method for survival analysis with numerous time-dependent covariates, offering a favorable balance between accuracy, variable selection, and computational speed.

Future work may explore the extension of the AACD framework to accommodate other types of penalization such as elastic-net or nonconvex penalties, which could further improve variable selection in highly correlated settings. Additionally, investigating adaptive strategies for dynamic time-dependent covariates, parallelized implementations for large-scale datasets, and integration with survival prediction ensembles may enhance both computational efficiency and predictive accuracy. Validation on diverse clinical and high-dimensional genomic datasets would also provide deeper insight into the generalization and practical utility of AACD.

Authors' Contributions

The authors equally contributed to this work. All authors read and approved the final manuscript.

Data Availability

For accessibility, we have made the data and R codes used in this study publicly available in the GitHub repository: `https://github.com/R-Ramezani/TDPCox-FUCOX`

Conflicts of Interest

The authors declare that there is no conflict of interest.

Ethical Considerations

The authors have diligently addressed ethical concerns, such as informed consent, plagiarism, data fabrication, misconduct, falsification, double publication, redundancy, submission, and other related matters.

Funding

Mohammad Arashi's work is based on the research supported in part by the Iran National Science Foundation (INSF) grant No. 4015320.

Using Artificial Intelligence Chatbots

We utilized AI-powered tools to check grammar and enhance the academic quality of the text, which was primarily written by the authors.

References

[1] P. K. Andersen and R. D. Gill, Cox's regression model for counting processes: A large sample study, The Annals of Statistics, 10(4), 1100–1120, (1982).

[2] J. Bradic, J. Fan and J. Jiang, Regularization for Cox's proportional hazards model with NP-dimensionality, The Annals of Statistics, 39(6), 3092–3120, (2011).

[3] N. Breslow, Covariance analysis of censored survival data, Biometrics, 30(1), 89–99, (1974).

[4] D. R. Cox, Regression models and life-tables, Journal of the Royal Statistical Society: Series B (Methodological), 34(2), 187–202, (1972).

[5] J. Fan and R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, Journal of the American Statistical Association, 96(456), 1348–1360, (2001).

[6] J. H. Friedman, T. Hastie and R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, Journal of Statistical Software, 33(1), 1–22, (2010).

[7] J. J. Goeman, L1 penalized estimation in the Cox proportional hazards model, Biometrical Journal, 52(1), 70–84, (2010).

[8] J. Gui and H. Li, Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data, Bioinformatics, 21(13), 3001–3008, (2005).

[9] F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee and R. A. Rosati, Evaluating the yield of medical tests, JAMA, 247(18), 2543–2546, (1982).

[10] P. J. Heagerty, T. Lumley and M. S. Pepe, Time-dependent ROC curves for censored survival data and a diagnostic marker, Biometrics, 56(2), 337–344, (2000).

[11] J. Huang, Y. Jiao, B. Jin, J. Liu, X. Lu and C. Yang, A unified primal–dual active set algorithm for nonconvex sparse recovery, Statistical Science, 36(2), 215–238, (2021).

[12] N. Simon, J. Friedman, T. Hastie and R. Tibshirani, Regularization paths for Cox's proportional hazards model via coordinate descent, Journal of Statistical Software, 39(5), 1–13, (2011).

[13] I. Sohn, J. Kim, S.-H. Jung and C. Park, Gradient lasso for Cox proportional hazards model, Bioinformatics, 25(14), 1775–1781, (2009).

[14] T. M. Therneau and T. Lumley, Package 'survival', R Package Version, 2.38, 1–37, (2015).

[15] T. M. Therneau, C. S. Crowson and E. Atkinson, Using time-dependent covariates and time-dependent coefficients in the Cox model, Survival Vignettes, 1, 1–25, (2017).

[16] R. Tibshirani, The LASSO method for variable selection in the Cox model, Statistics in Medicine, 16(4), 385–395, (1997).

[17] 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, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 74(2), 245–266, (2012).

[18] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, Journal of Optimization Theory and Applications, 109(3), 475–494, (2001).

[19] Y. Yang and H. Zou, A cocktail algorithm for solving the elastic net penalized Cox's regression in high dimensions, Statistics and Its Interface, 6(2), 167–173, (2013).

[20] H. Zou and T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301–320, (2005).