Greed Meets Sparsity: Understanding and Improving Greedy Coordinate Descent for Sparse Optimization

Huang Fang
University of British Columbia

Zhenan Fan
University of British Columbia

Yifan Sun
INRIA-Paris

Michael P. Friedlander
University of British Columbia

Abstract

We consider greedy coordinate descent (GCD) for composite problems with sparsity inducing regularizers, including 1-norm regularization and non-negative constraints. Empirical evidence strongly suggests that GCD, when initialized with the zero vector, has an implicit screening ability that usually selects at each iteration coordinates that at are nonzero at the solution. Thus, for problems with sparse solutions, GCD can converge significantly faster than randomized coordinate descent. We present an improved convergence analysis of GCD for sparse optimization, and a formal analysis of its screening properties. We also propose and analyze an improved selection rule with stronger ability to produce sparse iterates. Numerical experiments on both synthetic and real-world data support our analysis and the effectiveness of the proposed selection rule.

1 Introduction

Coordinate descent (CD) optimization algorithms operate on a single variable at each iteration, improving the objective value along a single coordinate while holding all other variables fixed. Both theoretical and empirical evidence point to the efficiency of this approach for large-scale optimization problems (Nesterov, 2012; Shalev-Shwartz and Zhang, 2013; Zhang and Lin, 2015; Wright, 2015; Allen-Zhu et al., 2016). Greedy coordinate descent (GCD) select at each iteration the coordinate that maximizes the marginal progress. It has been observed in practice that GCD, using the Gauss-Southwell coordinate selection rule (the GS rule), together with an all-zero initialization, may converge to a sparse solution significantly faster than CD methods that use a randomized selection rule (RCD), especially for high-dimensional problems. In particular, GCD applied to problems of the form LASSO (Hastie et al., 2008; Li and Osher, 2009; Nutini et al., 2015) and kernel support vector machines (SVMs) (Platt, 1999; Joachims, 1999; Chang and Lin, 2011) can sometimes get close to an optimal solution before executing even a single pass of all coordinates. This suggests that GCD exhibits an inherent screening ability for sparse optimization, preferring to update coordinates at which the final solution is nonzero.

We present a formal analysis to understand why GCD works well for problems with sparse solutions and an all-zero initialization. In addition, we propose and analyze an improved selection rule for GCD, and show its connection with existing algorithms. Experiments on both real world and synthetic data illustrate our analysis and the effectiveness of the approach.

2 Related Work

2.1 Coordinate descent (CD)

Coordinate descent methods for optimization have a long history, dating to 1940 (Southwell, 1940; Powell, 1973; Luo and Tseng, 1993; Bertsekas, 1999). Wright (2015) gives a comprehensive survey. Their key attraction is the low computational complexity of updating a single variable at each iteration, via either a complete minimization or a gradient update along the selected coordinate. These methods are prized for their efficiency on many machine learning problems, including LASSO (Hastie et al., 2008), SVMs (Platt, 1999), non-negative matrix factorization (Cichocki and Phan, 2009), and graph-based label propagation (Bengio et al., 2006). Nesterov's 2012 seminal work established the first non-asymptotic convergence rate for RCD, widely used in practice. Subsequent studies offered refined analyses on CD and its variants and explained why these methods are effective for many machine learning problems (Richtárik and Takác, 2014; Allen-Zhu et al., 2016; Nesterov and Stich, 2017).

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

2.2 Greedy coordinate descent (GCD)

A greedy-coordinate selection rule can greatly accelerate a CD method in practice. There are many such update rules, including the maximum-improvement rule, which picks a coordinate that allows maximal decrease; the GS rule, which picks the coordinate with the largest gradient magnitude, and a randomized-selection rule. The analyses offered by Nutini et al. (2015) and Karimireddy et al. (2019) explain why the GS rule is often faster than the randomized-selection rule, and describe a convergence rate, based on strong convexity with respect to the 1-norm, that is independent of problem dimension. Our approach builds primarily upon these last two references.

Implementation One obstacle for efficiently implementing the GS rule is its requirement for computing the maximal (in modulous) element of the full gradient. This operation generally requires a full gradient computation, which may be prohibitive for large problems. The GS rule is therefore most relevant for cases where the maximal-element may be found at a cost less than the cost of a full gradient computation. Nutini et al. (2015) describe a range of problem structures that allow for efficient maximal-element computation. Dhillon et al. (2011) and Karimireddy et al. (2019) describe a maximum inner-product search algorithm that approximates the GS rule. Other notable improvements to a standard CD implementation include parallel or distributed kernels suitable for multi-core or multi-machine environments (Liu et al., 2015; Richtárik and Takác, 2016). This line of work is tangential to our purpose and we do not discuss these further.

2.3 Sparsity pattern identification

A related line of work seeks to analyze the iteration complexity of identifying the sparsity pattern of an optimal solution (Dunn, 1987; Burke and Moré, 1988; Wright, 1993; Nutini et al., 2017b,a; Liang et al., 2017; Sun et al., 2019). The analysis approach used in this context is closely related to the screening properties studied in the present paper, and forms the basis for our approach.

3 Problem statement

We consider the optimization problem

$$\min_{x\in\mathbb{R}^d} \; F(x) := f(x) + g(x), \tag{1}$$

where d is the number of variables, f is a smooth and convex function, and $g(x) := \sum_{i=1}^d g_i(x_i)$ is a function that is separable and convex, but not necessarily smooth. We also make the following assumptions:

- $f(x+\alpha e_i)$ is L_i-smooth in terms of α $\forall i \in [d]$:

$$|\nabla_i f(x+de_i) - \nabla_i f(x)| \le L_i|d|,$$

for all $x \in \mathbb{R}^d$, where e_i is the ith unit vector. Let $L := \max_{i\in[d]} L_i$. Note that L can be much smaller than the gradient Lipschitz constant in $\mathbb{R}^d$.

- f is L_∞-smooth with respect to the ∞-norm:

$$\|\nabla f(x) - \nabla f(y)\|_\infty \le L_\infty \|x-y\|_1 \;\forall x,y\in\mathbb{R}^d.$$

- f is μ_p-strongly convex with respect to the p-norm:

$$f(x) \ge f(y) + \langle \nabla f(y), x-y\rangle + \frac{\mu_p}{2}\|x-y\|_p^2$$

for all $x, y \in \mathbb{R}^d$, where $p \in \{1,2\}$.

- $g = \lambda\|\cdot\|_1$, or $g = \delta_{\ge 0}$ is the function that vanishes on the nonnegative orthant, and is $+\infty$ otherwise. (We believe that it may be possible to relax this assumption and generalize our analysis to the case in which the constituent functions g_i are non-smooth at 0 for all $i \in [d]$.)

Algorithm 1 A generic template for GCD

Input: functions f and g_i $\forall i \in [d]$.
$W_0 = \emptyset$
$x^0 = 0$
for $t = 0, 1, 2, \ldots$ **do**
  Coordinate selection: select i according to rule
  Gradient step: $x^{t+\frac{1}{2}} = x^t - (1/L_i)\nabla f_i(x^t)e_i$
  Prox step: $x^{t+1} = \text{prox}_{(1/L_i)g_i}\left(x^{t+\frac{1}{2}}\right)$
  Optional post-processing; see (3)
  Update working set: $W_{t+1} = W_t \cup \{i\}$
end for

Algorithm 1 shows the generic template for the GCD method. We consider GCD under the following rule:

Selection rule 1 (GS-s rule)**.** *Select coordinate* $i \in \arg\max_{j\in[d]} Q_j(x^t)$*, where*

$$Q_j(x) = \min_{s\in\partial g_j} |\nabla_j f(x) + s|. \tag{2}$$

Although our subsequent theoretical development is based on the GS-s rule, it can be easily extended to include other selection rules described by Nutini et al. (2015); Bertsekas (1999); Tseng and Yun (2009); and Dhillon et al. (2011).

Karimireddy et al. (2019) propose an optional *post-processing* step useful to derive a convergence rate that depends on the strong convexity modulous μ_1 instead of μ_2, needed to remove the dependency on the dimension d. Specifically, after each prox-gradient step, set

$$x_i^{t+1} := 0 \;\text{ if }\; x_i^{t+1}x_i^t < 0. \tag{3}$$

We adopt this post-processing technique in the following sections. For simplicity, assume that there is only a

single element in the set $\arg\max_j Q_j(x_t)$, which makes Algorithm 1 deterministic. (In practice, the selection rule might break ties with a lexicographic ordering.)

Definition 1. Define the *working set* W_t as the set of indices selected up to and including iteration t. Define also $W := \bigcup_{t=0}^{\infty} W_t$ as the overall working set.

Definition 2. Define the *support* of a vector x as $\operatorname{supp}(x) = \{\, i \mid x_i \neq 0 \,\}$.

Definition 3. (Rockafellar, 1970, §23) For a closed, convex function $g : \mathbb{R}^d \to \mathbb{R}$, the subdifferential of g at x is defined as the set

$$\partial g(x) := \{\, v \in \mathbb{R}^d \mid g(y) \geq g(x) + v^T(y - x) \;\forall y \,\}$$

The set ∂g is always closed and convex. In particular, for $g_i : \mathbb{R} \to \mathbb{R}$ convex, the set ∂g_i is a closed interval in $\mathbb{R}$. Thus, for

- $g(x) = \lambda\|x\|_1$, $\partial g_i(0) = [-\lambda, \lambda]$;
- $g(x) = \delta_{\geq 0}(x)$, $\partial g_i(0) = (-\infty, 0]$.

3.1 The merits of keeping the iterates sparse

Karimireddy et al. (2019) derive the following linear convergence rate for GCD applied to strongly convex objectives with 1-norm regularization or non-negative constraints:

$$F(x^t) - F^* \leq \left(1 - \frac{\mu_1}{L}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F^*), \qquad (4)$$

where $F^* = \inf F$. Such rates illustrate the theoretical advantage of GCD over randomized CD, which exhibit the bound (4), except that the term μ_1/L is replaced with $\mu_2/(dL)$, where $\mu_1 \in [\mu_2/d, \mu_2]$ (Nutini et al., 2015). The rate improvement is especially salient when the dimension d is very large. A weakness of the rate (4) is its suggestion that GCD applied to strongly convex composite problems, with either 1-norm regularization or non-negative constraints, has the same rate as it does for problems without regularizers—i.e., minimizing only $f(x)$ instead of the sum $f(x) + g(x)$. However, GCD applied to the composite problem with sparsity inducing regularization is usually significantly faster than its non-sparse counterpart. An improved convergence analysis is thus needed to explain this phenomenon.

The following theorem sketch describes how sparse solutions translate into an improved convergence rate for GCD. See Theorems 4 and 5 for fuller descriptions.

Theorem Sketch 2 (GCD rate with sparsity)**.** *The t-th iterate x^t generated by Algorithm 1, using the GS-s selection rule (1), satisfies the bound*

$$F(x^t) - F^* \leq \left(1 - \frac{\tilde{\mu}}{L}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F^*), \qquad (5)$$

where the constant

$$\tilde{\mu} \in [\max\{\mu_2/|W|, \mu_1\}, \mu_2].$$

The linear convergence rate shown by this last result reveals the dependency of the rate constant on the sparsity of the solution: when the solution is very sparse—i.e., $|W| \ll d$—then the rate in bound (5) is much tighter than the rate predicted by (4), particularly when the constant $\mu_1 \approx \mu_2/d$.

Additionally, sparsity carries the benefit of allowing for sparse data structures to reduce memory requirements and faster matrix-vector multiplications.

4 Analysis

We now study the screening ability of GCD and develop a bound on the size of the working set W. We require the following quantity, often in sparsity pattern identification:

$$\delta_i := \min\{-\nabla_i f(x^*) - \ell_i,\; u_i + \nabla_i f(x^*)\}, \qquad (6)$$

where $\partial g_i(x_i^*) = [\ell_i, u_i]$ and $x^* = \arg\min F(x)$; see Hare and Lewis (2007); Lewis and Wright (2011); Nutini et al. (2017b); Sun et al. (2019). The constant δ_i is closely related to the distance to the relative interior of the sparse manifold, and is an important quantity in sparse manifold identification (Lewis and Wright, 2011). Optimality conditions for (1) imply that $\delta_i = 0$ if $x_i^* \neq 0$, and $\delta_i \geq 0$ if $x_i^* = 0$. Because x^* is unique, these quantities are problem-specific and algorithmically invariant. The definition in (6) leads to the following identification result.

Lemma 3 (Nutini et al. (2017b))**.** *If for some $t > 0$,*

$$|\nabla_i f(x^t) - \nabla_i f(x^*)| \leq \delta_i,$$

then after one coordinate proximal gradient step,

$$x_i^t = 0 \;\Rightarrow\; x_i^{t+1} = 0.$$

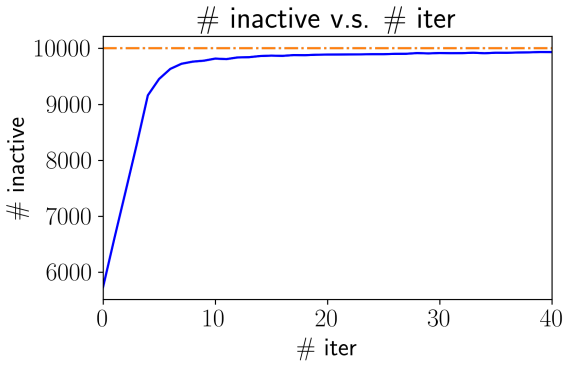
This lemma suggests that if $\nabla_i f(x^t)$ is close to $\nabla_i f(x^*)$ and $x_i^* = 0$, then the ith entry of x^t will be correctly identified as 0.

4.1 Numerical motivation

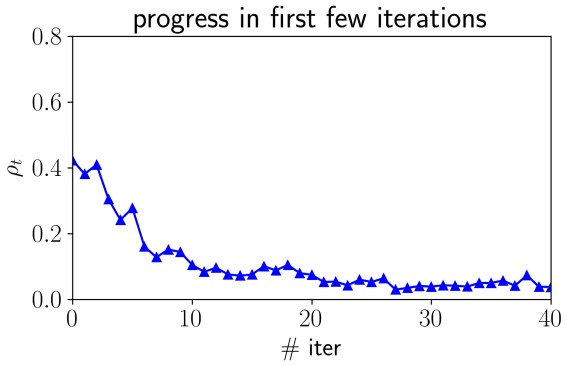
Our analysis approach is based on the following numerical observations, which we illustrate with an example LASSO problem on random synthetic data. Define

$$f(x) = \tfrac{1}{2}\|Ax - b\|_2^2 \quad \text{and} \quad g(x) = \lambda\|x\|_1, \qquad (7)$$

where $A \in \mathbb{R}^{50\times 10^4}$ and $b := Ax^\sharp + \epsilon$. The elements A_{ij}, ϵ_i, and nonzeros in the solution $x^\sharp$ are distributed as standard Gaussians. We randomly select 10 elements of $x^\sharp$ to be nonzero and set $\lambda = 2$.



(a) Number of inactive variables plot.



(b) Objective progress, where ρ_t is defined in (9).

Figure 1: Exploratory investigations.

Figures 1a and 1b show the evolution of the number of "inactive" variables and objective progress for Algorithm 1.

In Figure 1a, we define

$$\#\,\text{inactive} := \sum_{i=1}^{d} \mathbf{1}\left\{|\nabla_i f(x^t) - \nabla_i f(x^*)| \le \delta_i\right\}. \quad (8)$$

According to Lemma 3, this quantity measures how many variables are staying "inactive", i.e., do not move away from 0 in the next iteration. From Figure 1a, we find that most variables are initially incorrectly labeled as "active", i.e., $|\nabla_i f(x^t) - \nabla_i f(x^*)| > \delta_i$, but a large number of them quickly switch to "inactive".

In Figure 1b, we illustrate the objective progress at each step by plotting ρ_t, defined to satisfy

$$F(x^{t+1}) - F^* = (1-\rho_t)\left(F(x^t) - F^*\right). \quad (9)$$

These experiments illustrate the fact that initial convergence of GCD, which, for sparse solutions, may be sufficient to quickly identify the few nonzeros. From this experiment we observe that

- GCD converges fast initially and $\nabla f(x^t)$ quickly approaches $\nabla f(x^*)$ when x^t is still sparse; and
- before $|\text{supp}(x^t)|$ has grown significantly, the coordinates i where $x_i^* = 0$ have mostly become inactive, and thus future coordinates that enter W are constrained to $\text{supp}(x^t)$.

We now rigorously characterize these observations. Before proceeding to our results, we introduce some needed concepts.

Definition 4. The function f is $\mu_p^{(\tau)}$ strongly convex with respect to $\|\cdot\|_p$ and sparse vectors if $\forall x, y \in \mathbb{R}^d$ such that whenever $|\text{supp}(x) \cup \text{supp}(y)| \le \tau$,

$$f(x) \ge f(y) + \langle \nabla f(y), x - y\rangle + \frac{\mu_p^{(\tau)}}{2}\|x-y\|_p^2,$$

where $p \in \{1, 2\}$.

It can be easily verified that $\mu_1^{(\tau)}$ and $\mu_2^{(\tau)}$ satisfy the following conditions:

$$\begin{aligned} &\mu_1 = \mu_1^{(d)} \le \mu_1^{(d-1)} \le \cdots \le \mu_1^{(1)}, \\ &\mu_2 = \mu_2^{(d)} \le \mu_2^{(d-1)} \le \cdots \le \mu_2^{(1)}, \\ &\mu_2^{(\tau)}/\tau \le \mu_1^{(\tau)} \le \mu_2^{(\tau)} \ \forall \tau \in [d]. \end{aligned} \quad (10)$$

Next, we present a formal analysis to answer why GCD may converge fast initially, and give a bound on the size of the working set W.

4.2 Fast initial convergence

Theorem 4. *Let $\tau = |\text{supp}(x^*)|$ and let $\{x^i\}_{i=1}^{\infty}$ be the iterates generated by Algorithm 1 with the GS-s rule (selection rule 1). Then for $t < d - \tau$,*

$$F(x^t) - F^* \le \prod_{i=1}^{\lceil \frac{t}{2} \rceil}\left(1 - \frac{\mu_1^{(\tau+i-1)}}{L}\right)(F(0) - F^*) \quad (11)$$

$$\le \prod_{i=1}^{\lceil \frac{t}{2} \rceil}\left(1 - \frac{\mu_2}{(\tau+i-1)L}\right)(F(0) - F^*). \quad (12)$$

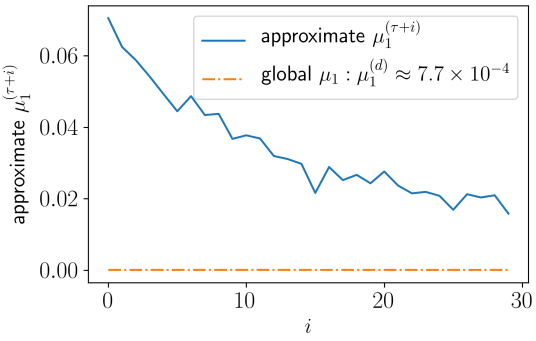
The bound in (12) follows from (10).

In Theorem 4 we show two different bounds, which allow us to draw comparisons to existing results, below.

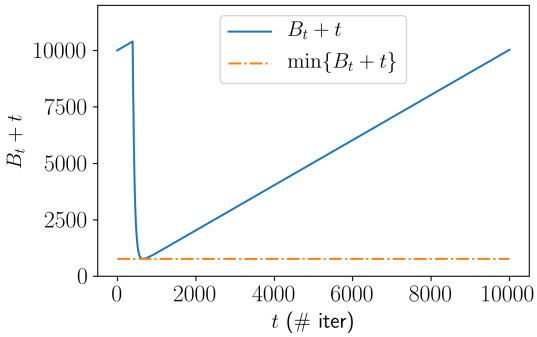
Bound (12). Nesterov (2012) and Richtárik and Takác (2014) establish that RCD exhibits the rate

$$\mathbb{E}\left[F(x^t) - F(x^*)\right] \le \left(1 - \frac{\mu_2}{dL}\right)(F(0) - F^*).$$

Compared to (12), we see that the dimension d is replaced by the quantity $(\tau+i-1)$, which, if $\tau = \text{supp}(x^*)$ is small and we are in the first few iterations, may be much smaller than d. This reflects the fast initial convergence often observed in practice; cf. Figure 1b.



(a) The trend of approximate $\mu_1^{(\tau+i)}$, where $\tau = 10$ in this example.



(b) The curve illustrates the upper bound of $|W|$ in Theorem 5.

Figure 2: Illustrations for Theorem 4 and 5

Bound (11). Nutini et al. (2015) and Karimireddy et al. (2019) establish for the GCD method the linear convergence rate described by (4). Compared to (11), we see that μ_1 is replaced with the quantity $\mu_1^{(\tau+i-1)}$, which is potentially much larger in the early stages (i small), particularly if τ is small (x^* is sparse). This is confirmed by Figure 2a, which shows how in practice this quantity can be much larger than μ_1. In particular, when τ and i are small, $\mu_2/d \ll \mu_1^{(\tau+i-1)}$ and the convergence rate for GCD is initially significantly faster than RCD, even in the worst case.

The rate we derived here is based on two important ingredients: zero initialization and sparse solution. The screening ability of GCD does not hold without either of these properties.

To better understand the effect of the quantity $\mu_1^{(\tau+i-1)}$ on the convergence rate, we conduct a simulation using the LASSO problem in (7). The term $\mu_1^{(\tau+i-1)}$ is hard to compute in general, and thus here we set $\tau = 10$, and for each $i \in \{1, 2, \ldots, 30\}$ we generate 10^3 random $(\tau + i - 1)$-sparse vectors and approximate $\mu_1^{(\tau+i-1)}$ as the minimum of $\|Ax\|_2^2/\|x\|_1^2$ over the sample vectors. The plot of approximate $\mu_1^{(\tau+i-1)}$ against i is shown in Figure 2a, and its pattern clearly supports our previous argument.

4.3 Fast support identification

In this section we define bounds on the size of the working set W. Define the error measure

$$p_\delta(\alpha) = \sum_{i=1}^{d} \mathbf{1}\{\alpha \le \delta_i\},$$

which we use to quantify the number of inactive elements in the iterates, as in (8).

Theorem 5 (Working set bound). *Let $\{x^i\}_{i=1}^\infty$ be the iterates generated by Algorithm 1 with the GS-s rule (selection rule 1). Then*

$$|W| \le \min_{t \in [d]} \{B_t + t\}, \tag{13}$$

where

$$B_t := d - p_\delta\left(L_\infty \sup_{i \ge t} \{\|x^i - x^*\|_1\}\right).$$

In order to better understand this bound, note that B_t is a decreasing function of t. Thus, $|W|$ is bounded by the infimum of the sum of a decreasing and increasing function. (See Figure 2b.) Theorem 5 implies that if x^t converges quickly to x^* (i.e., with $t \ll d$), then the bound (13) will be far less than d.

Again, consider the synthetic LASSO problem ($A \in \mathbb{R}^{50\times 10^4}, \lambda = 2$) as a concrete example to illustrate this bound. In this example, the curve of $B_t + t$ is shown in Figure 2b and the infimum of $B_t + t$ is about 1000 in this case. This experiment demonstrates that the bound we derived in Theorem 5 is non-trivial, especially for problems where d is large.

We use Theorems 4 and 5 to derive an alternative bound that depends only on the constant $\mu_1^{(\tau+i)}, i \in [d - \tau]$, instead of the iterates x^i.

Corollary 6. *Let $\tau = |\mathrm{supp}(x^*)|$ and let $\{x^i\}_{i=1}^\infty$ be the sequence of iterates generated by Algorithm 1 with the GS-s rule (selection rule 1). Then B_t in bound (13) can be replaced by*

$$B_t := d - p_\delta\left(\left[\frac{2L_\infty^2}{\mu_1}\prod_{i=0}^{t-1}\left(1 - \frac{\mu_1^{(\tau+i)}}{L}\right)R\right]^{1/2}\right),$$

where $R = F(0) - F^*$ is the initial objective gap.

5 Improved selection rule

Our analysis in section 4 provides a bound on the size of the working set W, and thus provides an explanation

for why GCD is fast for sparse optimization. But GCD could be potentially slow in some situations, for example when $|W|$ is large. Can we improve GCD by trying to keep $|W|$ small? In answer, we propose a variant of the GS-s selection rule that favours a small final working set. The resulting algorithm, which we call Δ-GCD, is Algorithm 1 with the following modified selection rule.

Selection rule 7 (Δ-GS-s rule)**.** *Given the fixed parameter $\Delta \in (0, 1]$, select coordinate*

$$i \in \begin{cases} \arg\max_{i\in[d]} Q_i(x^t), & \Delta \max_{i\in[d]} Q_i(x^t)^2 \geq \max_{i\in W_t} Q_i(x^t)^2 \\ \arg\max_{i\in W_t} Q_i(x^t), & \Delta \max_{i\in[d]} Q_i(x^t)^2 < \max_{i\in W_t} Q_i(x^t)^2 \end{cases}$$

where W_t denotes the set of indices accrued thus far and Q_i is defined by (2).

Note that when $\Delta = 1$, the Δ-GS-s and GS-s rules are equivalent.

Intuitively, the Δ-GS-s rule, with small Δ, is more likely to focus on the current working set; on the other hand, a large Δ encourages the algorithm to explore new coordinates and expand the current working set. Thus Δ controls the trade-off between the size of working set and the progress we made when staying in the current working set. This is similar to the exploration/exploitation trade-off in the context of online learning (Auer et al., 1995).

Theorem 8. *Let $\{x^i\}_{i=1}^{\infty}$ be the iterates generated by Algorithm 1 with the Δ-GS-s rule (selection rule 7) and let W_Δ be the final working set. Then for all $t > 0$,*

$$F(x^t) - F^* \leq \left(1 - \frac{\Delta\mu_1^{(|W_\Delta|)}}{L}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F^*) \quad (14)$$

$$\leq \left(1 - \frac{\Delta\mu_2}{|W_\Delta|L}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F^*). \quad (15)$$

Theorem 8 makes explicit the trade-off between the convergence rate and the size of working set. Again, we provide two bounds for easier interpretation: (14) as a refinement of the strong convexity parameter in Karimireddy et al. (2019) and Nutini et al. (2015); and (15) where the variable dimension dependency that appears in Nesterov and Stich (2017) and Richtárik and Takác (2014) is replaced by the size of the final working set. The Δ-GCD variant is expected to outperform standard GCD when the latter has a comparatively large working set, and Δ-GCD can reduce the size of working set with an appropriate value of Δ.

To better understand the relationship between W_Δ and Δ, we present an description of W_Δ as $\Delta \to 0$. First, consider the standard GCD algorithm where at each iteration we additionally minimize the objective over the current working set, i.e., the next iterate is obtained as

$$x^{t+1} := \arg\min_{\text{supp}(x)\subseteq W_{t+1}} f(x) + g(x), \quad (16)$$

where all variables not in the working set are held fixed at 0. The algorithm terminates when the iterate x^{t+1} is optimal for (1). The resulting method is known as the *totally corrective greedy algorithm*, which is closely related to orthogonal matching pursuit for sparse least squares; see Pati et al. (1993); Davis et al. (1997); and Foucart and Rauhut (2013, §3.2). We denote the final working set from this scheme as $W^\sharp$.

Next, note that as $\Delta \to 0$, the Δ-GS-s selection rule tends to select indices from the current working set, and thus the Δ-GS algorithm converges to a solution of (16). However, when the Δ-GCD iterate is close to the exact minimizer, the Δ-GS-s rule must eventually expand the workset. As the following result shows, W_Δ converges to $W^\sharp$.

Theorem 9. *Let $k = |W^\sharp|$ and let $\{x^t\}_{i=0}^k$ be the iterates generated by the totally corrective greedy algorithm. Assume that $\arg\max_{i\in[d]} Q_i(x^t)$ are singletons for $t = 0, 1, \ldots, k-1$ and $\delta_i > 0$ $\forall x_i^* = 0$. Then*

$$\lim_{\Delta\to 0} W_\Delta = W^\sharp.$$

If the totally corrective greedy algorithm can yield a small working set, then we expect that a sufficiently small value of Δ would also yield a small working set (but could probably slow down the convergence according to Theorem 8). Hence, our new algorithm Δ-GCD can be viewed as a flexible greedy algorithm between the two extreme cases—standard GS-GCD and the totally corrective greedy algorithm.

Table 1: Properties of the experimental data. Here, d denotes the number of features and n denotes the number of samples.

Datasets	colon	leukemia	make_circle	ijcnn1
d	2,000	7,129	2	22
n	62	72	1,000	35,000

6 Experiments

In this section, we describe experiments on both real world data and synthetic data to illustrate the importance of zero initialization and evaluate the effectiveness of Δ-GCD.

The statistics of our experimental data are shown in Table 1, where the datasets colon, leukemia, and ijcnn1 were obtained from the LIBSVM website (Chang

and Lin, 2011). The make_circle dataset were generated from the scikit-learn package (Pedregosa et al., 2011). We solve the LASSO problem over the colon and leukemia datasets, and the dual RBF kernel SVM over the make_circle and ijcnn1 datasets. For ijcnn1, we follow the parameter settings described by Hsieh et al. (2014), and thus set $\gamma = 2$ and $C = 32$, where γ is the free parameter in the RBF kernel and $1/C$ is the hinge-loss weight parameter. All experiments are conducted on a machine with 4 CPUs and 16GB memory.

The code to reproduce our experimental results is publicly available at https://github.com/fanghgit/Greed_Meets_Sparsity.

6.1 Zero v.s. other initializations

In this section, we compare the convergence of standard GCD for solving LASSO problems over different initializations.

- Zero initialization: $x^0 = 0$.
- Random initialization: x^0 is generated from Gaussian distributions $\mathcal{N}(0, \sigma I_d)$, for $\sigma \in \{1, 0.1, 0.01\}$.
- Least-squares initialization: $x^0 = (A^T A + \lambda I_d)^{-1} A^T b$. This initialization starts the method at a low objective value, but is not sparse.

In Figure 4, we can see that zero initialization clearly outperforms other initialization strategies, and random initialization tends to be the worst. In particular, GCD with zero initialization can get close to a solution even before one pass of all coordinates. On the other hand, although GCD with least-squares initialization has a smaller initial objective value than zero initialization, it suffers from slow convergence and requires a full pass of all coordinates before reaching the same low error, which is consistent with our intuition. Random initialization with different standard deviation also varies in their performance and random initialization, with smaller variance tends to converge faster; however they are still outperformed by the zero initialization for the same reasons as the least-squares initialization.

6.2 Evaluation of Δ-GCD

In this section, we evaluate the proposed Δ-GCD algorithm on LASSO, 1-norm regularized logistic regression, and kernel SVM problems. As shown in Figure 5, the value of Δ has a clear impact on the size of the working set, where smaller values of Δ tend to promote sparser iterates for all the test problems. This trend is more obvious when the underlying solution is less sparse, as shown in Figures 5b and 5f. This is because vanilla GCD produces a working set that is much larger than needed. Sometimes this stronger screening ability of a smaller Δ can lead to slightly faster convergence compared to standard GCD (i.e., $\Delta = 1$) as shown in Figures 5a and 5b. A by-product of Δ-GCD is early identification of the final sparsity pattern, which can be leveraged in two-stage methods (Bertsekas, 1976; Ko et al., 1994; Daniilidis et al., 2009; Wright, 2012). However, the acceleration functionality of Δ-GCD is not present for all test problems, since vanilla GCD already has a very strong screening ability for constraining the size of the working set.

Figure 3 shows the size of the working set after 10^5 iterations (as an approximation for the true $|W_\Delta|$) with different choices of $\Delta \in \{2^{-k} \mid k = 0, 1, \ldots, 6\}$ on a LASSO problem with dataset colon and $\lambda = 0.1$. As shown in the figure, the size of $|W_\Delta|$ is monotonically decreasing with Δ.

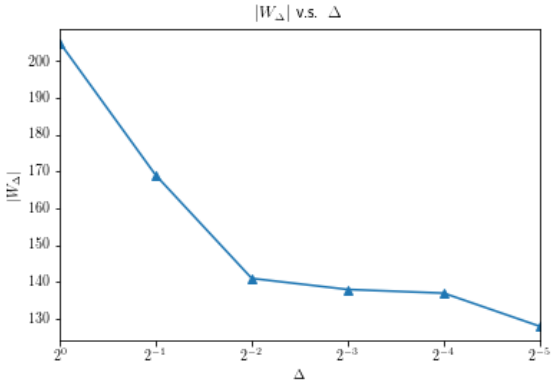
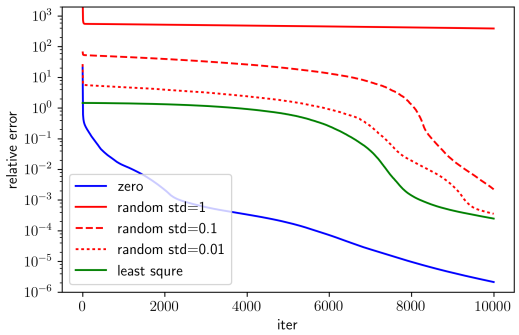


Figure 3: The effect of Δ on W_Δ.

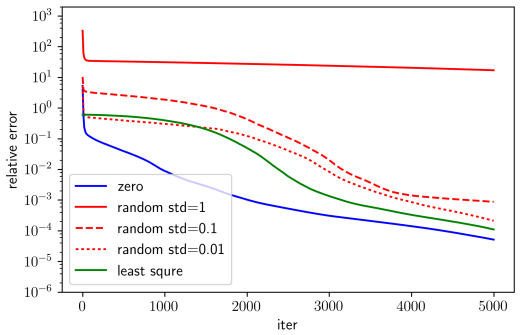
7 Conclusions

By bringing techniques from sparsity pattern identification and convergence analysis of GCD, we formally analyze the screening ability of GCD and explicitly answered why GCD is usually fast for sparse optimization. We also propose an improved selection rule with a stronger ability to encourage sparse iterates and connect to existing algorithms.

For future work, we would like to generalize our analysis and relax the strong-convex assumption on the function f. In particular, we wish to consider problems where x^* may not be unique (but $\text{supp}(x^*)$ may be). The core of our analysis relies on understanding the convergence of the iterates themselves, and not just the function values. Thus, the challenge in generalizing our analysis to more general smooth objectives requires a different proof technique. We also wish to consider to tighten the working set bound in Theorem 5. The bound illustrated in Figure 2b is still about 10 times worse than the actual size of the working set, and for small value of λ, the actual working set become larger and our bound can be trivially larger than d.

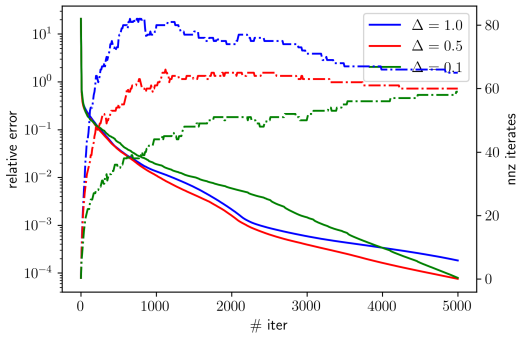


(a) LASSO, data: leukemia, $\lambda = 0.1$.

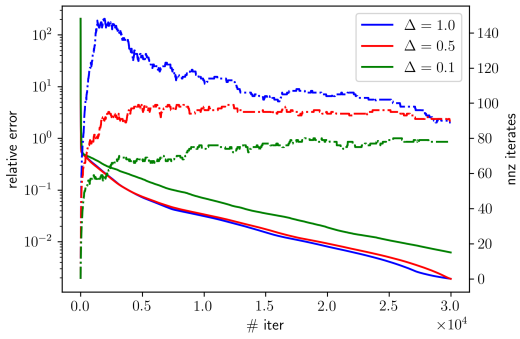


(b) LASSO, data: colon, $\lambda = 0.1$.

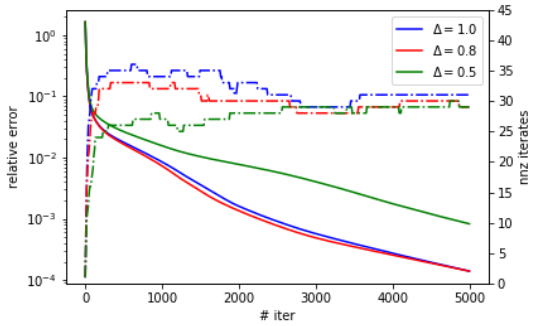
Figure 4: Comparison between different kinds of initialization



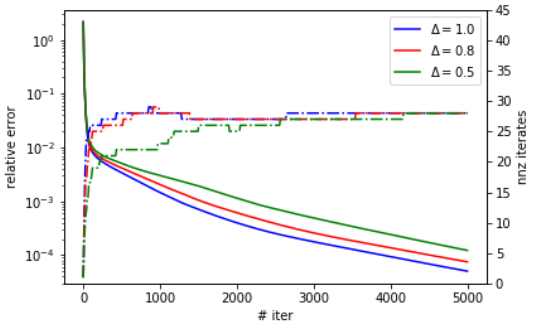
(a) LASSO, data: leukemia, $\lambda = 0.1$. Solid = obj. value, dashed = # nonzeros in x^t.



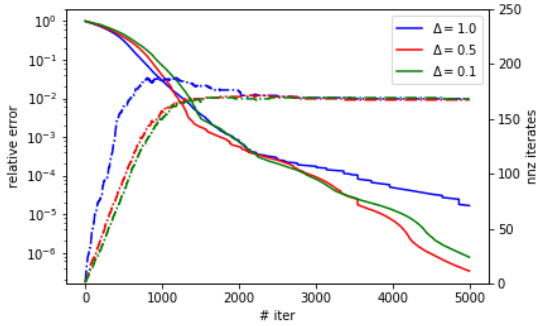
(b) LASSO, data: leukemia, $\lambda = 0.01$. Solid = obj. value, dashed = # nonzeros in x^t.



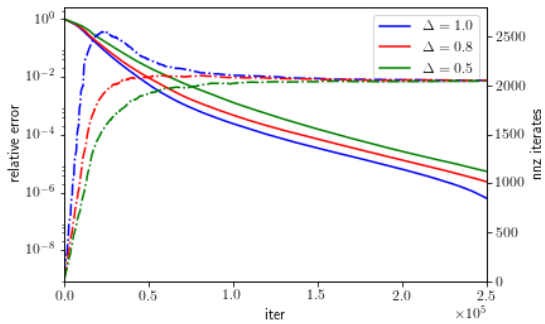
(c) L_1-regularized logistic regression, data: colon, $\lambda = 0.1$.



(d) L_1-regularized logistic regression, data: leukemia, $\lambda = 0.1$.



(e) kernel SVM, data: make_circle, $C = 10, \gamma = 0.5$.



(f) kernel SVM, data: ijcnn1, $C = 32, \gamma = 2$

Figure 5: Compare Δ-GCD with different choices of Δ.

References

Allen-Zhu, Z., Qu, Z., Richtarik, P., and Yuan, Y. (2016). Even faster accelerated coordinate descent using non-uniform sampling. In *Proceedings of ICML*, volume 48, pages 1110–1119.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (1995). Gambling in a rigged casino: The adversarial multi-arm bandit problem. In *Proceedings of FOCS*, pages 322–331.

Bengio, Y., Delalleau, O., and Le Roux, N. (2006). Label propagation and quadratic criterion. In *Semi-Supervised Learning*, pages 193–216. MIT Press.

Bertsekas, D. P. (1976). On the goldstein-levitin-polyak gradient projection method. *IEEE Transactions on automatic control*, 21(2):174–184.

Bertsekas, D. P. (1999). *Nonlinear Programming*, volume second edition. Athena Scientific.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Burke, J. V. and Moré, J. J. (1988). On the identification of active constraints. *SIAM Journal on Numerical Analysis*, 25(5):1197–1211.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Cichocki, A. and Phan, A. H. (2009). Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions*, 92-A(3):708–721.

Daniilidis, A., Sagastizábal, C., and Solodov, M. (2009). Identifying structure of nonsmooth convex functions by the bundle technique. *SIAM Journal on Optimization*, 20(2):820–840.

Davis, G., Mallat, S., and Avellaneda, M. (1997). Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98.

Dhillon, I. S., Ravikumar, P., and Tewari, A. (2011). Nearest neighbor based greedy coordinate descent. In *Proceedings of NIPS*, pages 2160–2168.

Dunn, J. C. (1987). On the convergence of projected gradient processes to singular critical points. *Journal of Optimization Theory and Applications*, 55(2):203–216.

Foucart, S. and Rauhut, H. (2013). *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel.

Hare, W. L. and Lewis, A. S. (2007). Identifying active manifolds. *Algorithmic Operations Research*, 2(2):75.

Hastie, T., Friedman, J. H., and Tibshirani, R. (2008). Regularization paths and coordinate descent. In *Proceedings of ACM-SIGKDD*, page 3.

Hsieh, C.-J., Si, S., and Dhillon, I. (2014). A divide-and-conquer solver for kernel support vector machines. In *Proceedings of ICML*, volume 32 of *Proceedings of Machine Learning Research*, pages 566–574. PMLR.

Joachims, T. (1999). Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA.

Karimireddy, S. P., Koloskova, A., Stich, S. U., and Jaggi, M. (2019). Efficient greedy coordinate descent for composite problems. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 2887–2896. PMLR.

Ko, M., Zowe, J., et al. (1994). An iterative two-step algorithm for linear complementarity problems. *Numerische Mathematik*, 68(1):95–106.

Lewis, A. S. and Wright, S. J. (2011). Identifying activity. *SIAM Journal on Optimization*, 21(2):597–614.

Li, Y. and Osher, S. (2009). Coordinate descent optimization for l_1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Probl. Imaging*, 3:487–503.

Liang, J., Fadili, J., and Peyré, G. (2017). Activity identification and local linear convergence of forward-backward-type methods. *SIAM Journal on Optimization*, 27(1):408–437.

Liu, J., Wright, S. J., Ré, C., Bittorf, V., and Sridhar, S. (2015). An asynchronous parallel stochastic coordinate descent algorithm. *J. Mach. Learn. Res.*, 16:285–322.

Luo, Z. and Tseng, P. (1993). Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46-47(1):157–178.

Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362.

Nesterov, Y. and Stich, S. U. (2017). Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123.

Nutini, J., Laradji, I., and Schmidt, M. (2017a). Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence.

Nutini, J., Schmidt, M., and Hare, W. (2017b). "active-set complexity" of proximal gradient: How long does

it take to find the sparsity pattern? *Optimization Letter*.

Nutini, J., Schmidt, M. W., Laradji, I. H., Friedlander, M. P., and Koepke, H. A. (2015). Coordinate descent converges faster with the gauss-southwell rule than random selection. In *Proceedings of ICML*, pages 1632–1641.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods*, pages 185–208. MIT Press.

Powell, M. J. D. (1973). On search directions for minimization algorithms. *Math. Program.*, 4(1):193–201.

Richtárik, P. and Takác, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math. Program.*, 144(1-2):1–38.

Richtárik, P. and Takác, M. (2016). Parallel coordinate descent methods for big data optimization. *Math. Program.*, 156(1-2):433–484.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press, Princeton.

Shalev-Shwartz, S. and Zhang, T. (2013). Stochastic dual coordinate ascent methods for regularized loss minimization. *J. Mach. Learn. Res.*, 14(1):2013.

Southwell, R. V. (1940). Relaxation methods in engineering science : a treatise on approximate computation.

Sun, Y., Jeong, H., Nutini, J., and Schmidt, M. W. (2019). Are we there yet? manifold identification of gradient-related proximal methods. In *Proceedings of AISTATS*, pages 1110–1119.

Tseng, P. and Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program.*, 117(1-2):387–423.

Wright, S. J. (1993). Identifiable surfaces in constrained optimization. *SIAM Journal on Control and Optimization*, 31(4):1063–1079.

Wright, S. J. (2012). Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization*, 22(1):159–186.

Wright, S. J. (2015). Coordinate descent algorithms. *Math. Program.*, 151(1):3–34.

Zhang, Y. and Lin, X. (2015). Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proceedings of ICML*.

Supplement

A Preliminaries

We introduce some notations and Lemmas that appear in Nutini et al. (2015) and Karimireddy et al. (2019).

We say a gradient step is *good* if the post-processing step in (3) is not triggered i.e., $x_i^{t+1}x_i^t \geq 0$, otherwise we call this step a *bad* step. We denote the set of good steps until the t-th iteration as $\mathbb{G}_t$, since a bad step always follows a good step, it is easy to verify that

$$|\mathbb{G}_t| \leq \left\lceil \frac{t}{2} \right\rceil. \tag{17}$$

Recall the selection rule in section 3:

Selection rule 10 (GS-s rule)**.** *Select* $i \in \arg\max_j Q_j(x^t)$ *where*

$$Q_i(x) = \min_{s \in \partial g_i} |\nabla_i f(x) + s|. \tag{18}$$

Lemma 11 (Karimireddy et al. (2019))**.** *Assume* $f(\cdot)$ *is* μ_1 *strongly convex with respect to 1-norm, then the iterates generated from Algorithm 1 with GS-s rule (selection rule 2) satisfy*

$$F(x^t) - F(x^*) \leq \left(1 - \frac{\mu_1}{L}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F(x^*)).$$

The above lemma is from Karimireddy et al. (2019).

Lemma 12 (Karimireddy et al. (2019))**.** *Consider* $g(\cdot)$ *to be* ℓ_1 *regularization or non-negative constraint. Then if the* t*-th iteration is a good step, we have*

$$F(x^{t+1}) \leq F(x^t) - \frac{1}{2L} \max_{i \in [d]} Q_i(x^t)^2, \tag{19}$$

where $Q_i(\cdot)$ *is defined in the GS-s rule (selection rule 2).*

B Proof of Theorem Sketch 2

Let $W = \{w_1, w_2, ..., w_k\}$ s.t. $w_1 < w_2 < ... < w_k \in \mathbf{N}$, we define new functions $h(\cdot) : \mathbb{R}^k \to \mathbb{R}, h(y) = f(\sum_{i=1}^k y_i e_{w_i})$ and $H(y) := h(y) + \sum_{i=1}^k g_{w_i}(y_i)$.

First, we show that $h(y + \alpha e_i)$ is also L-smooth $\forall i \in [k]$.

For any $i \in [k], y \in \mathbb{R}^k$,

$$\begin{aligned} h(y + \alpha e_i) &= f(\sum_{j=1}^k y_j e_{w_j} + \alpha e_{w_i}) \\ &\leq f(\sum_{j=1}^k y_j e_{w_j}) + \alpha \nabla_{w_i} f(\sum_{j=1}^k y_j e_{w_j}) + \frac{L}{2}\alpha^2 \\ &= h(y) + \alpha \nabla_i h(y) + \frac{L}{2}\alpha^2 \end{aligned} \tag{20}$$

Second, we show that we can get the same iterates if we run GCD on $F(x)$ or $H(y)$, that is, we want to show that $x^t = \sum_{i=1}^k e_{w_i} y_i^t \ \forall t \geq 0$. We prove by induction:

When $t = 0$, obviously we have $x^0 = \sum_{i=1}^k e_{w_i} y_i^0 = 0$.

Suppose that $x^t = \sum_{i=1}^k e_{w_i} y_i^t$, $i = \arg\max_j Q_j(x^t)$, $\tilde{i} = \arg\max_j Q_j(y^t)$ and $i = w_m$, we can show that $\tilde{i} = m$:

Note that

$$\begin{aligned} Q_i(x^t) &= \min_{s\in\partial g_i} |\nabla_i f(x^t) + s| \\ &= \min_{s\in\partial g_m} |\nabla_m h(x) + s| \\ &= Q_m(y^t), \end{aligned} \tag{21}$$

Thus, it is easy to see that $\tilde{i} = m$.

$$x_i^{t+\frac{1}{2}} = x_i^t - \frac{1}{L}\nabla f_i(x^t) = y_m^t - \frac{1}{L}\nabla h_m(y^t) = y_m^{t+\frac{1}{2}}.$$

Note that $g_i(\cdot) = g_{w_m}(\cdot)$, thus we further have

$$x_i^{t+1} = \text{prox}_{\frac{1}{L}g_i}\left[x_i^{t+\frac{1}{2}}\right] = \text{prox}_{\frac{1}{L}g_{w_m}}\left[y_m^{t+\frac{1}{2}}\right] = y_{m+1}^{t+1}.$$

Thus we have $x^t = \sum_{i=1}^k e_{w_i} y_i^t \ \forall t = 0, 1, 2, ...$

Plug $H(\cdot)$ into Lemma 11 and using the above result, we can get

$$F(x^t) - F(x^*) = H(y^t) - H(y^*) \le \left(1 - \frac{\tilde{\mu}_1}{L}\right)^{\left\lceil \frac{t}{2} \right\rceil} (H(0) - H(y^*)) = \left(1 - \frac{\tilde{\mu}_1}{L}\right)^{\left\lceil \frac{t}{2} \right\rceil} (F(0) - F(x^*)),$$

where $\tilde{\mu}_1$ is the 1-norm strongly convex constant for the k-dimensional small problem $H(\cdot)$, since H is also μ_2 strongly convex, we can easily verify that $\max\{\mu_2/k, \mu_1\} \le \tilde{\mu}_1 \le \mu_2$, which completes the proof.

C Proof of Lemma 3

If i is not select by Algorithm 1 at the t-th iteration, then $x_i^{t+1} = 0$ trivially remains 0.

If i is selected at the t-th iteration, by assuming $|\nabla_i f(x^t) - \nabla_i f(x^*)| \le \delta_i$, we know that

$$\begin{aligned} &-\delta_i + \nabla_i f(x^*) \le \nabla_i f(x^t) \le \delta_i + \nabla_i f(x^*) \\ \overset{\text{(i)}}{\Rightarrow} &-u_i \le \nabla_i f(x^t) \le -l_i, \end{aligned} \tag{22}$$

where (i) follows directly from the definition of $\delta_i := \min\{-\nabla_i f(x^*) - l_i, u_i + \nabla_i f(x^*)\}$.

Then we show that $\text{prox}_{\frac{g}{L_i}}(0 - \frac{1}{L_i}\nabla_i f(x^t)) = 0$:

$$\text{prox}_{\frac{g}{L_i}}\left(0 - \frac{1}{L_i}\nabla_i f(x^t)\right) = \arg\min_y \left\{\frac{1}{2}\left(y - \left(-\frac{1}{L_i}\nabla f_i(x^t)\right)\right)^2 + \frac{1}{L_i}g_i(y)\right\} \tag{23}$$

This minimization problem is strongly convex and thus has a unique solution satisfies:

$$0 \in y + \frac{1}{L_i}\nabla_i f(x^t) + \frac{1}{L_i}\partial g_i(y) \tag{24}$$

By knowing $-u_i \le \nabla_i f(x^t) \le -l_i$ from (22) and $\text{int}\partial g_i(0) = (l_i, u_i)$ by the definition of l_i and u_i. We can easily conclude that $y = 0$ satisfies (24) and therefore

$$x_i^{t+1} = \text{prox}_{\frac{g}{L_i}}\left(0 - \frac{1}{L_i}\nabla_i f(x^t)\right) = 0.$$

D Proof of Theorem 4

Let $t \leq d - \tau$ and recall the definition of *good* steps until the t-th iteration from section A in Appendix: $|\mathbb{G}_t| = \{i_1, i_2, \ldots, i_k\}$, where $k \geq \lceil \frac{t}{2} \rceil$.

At iteration $i_m, m \in [k]$, x^{i_m} is guaranteed to be $m-1$–sparse, by assuming $f(\cdot)$ is $\mu_1^{(\tau+m-1)}$ strongly convex w.r.t. 1-norm and $\tau + m - 1$–sparse vectors, we know that $F(\cdot)$ is also μ_1 strongly convex w.r.t. 1-norm and $\tau + m - 1$–sparse vectors. Thus $\forall y \in \mathbb{R}^d$ that is τ-sparse, $|\mathrm{supp}(y) \cup \mathrm{supp}(x^{i_m})| \leq \tau + m - 1$ and by the definition of $\mu_1^{(\tau+m-1)}$, we have

$$F(y) \geq F(x^{i_m}) + \langle \partial F(x^{i_m}), y - x^{i_m} \rangle + \frac{\mu_1^{(\tau+m-1)}}{2} \|y - x^{i_m}\|_1^2, \tag{25}$$

with a little bit abuse of notation, here $\partial F(x^t)$ stands for any vector in the subdifferential of $F(x^t)$. Taking minimum on both side of (25) w.r.t. y that is τ sparse,

$$\begin{aligned}
F(x^*) &\geq F(x^{i_m}) - \sup_{\|y\|_0 \leq \tau} \left(\langle -\partial F(x^{i_m}), y - x^{i_m} \rangle - \frac{\mu_1^{(\tau+m-1)}}{2} \|y - x^{i_m}\|_1^2 \right) \\
&\geq F(x^{i_m}) - \sup_{y \in \mathbb{R}^d} \left(\langle -\partial F(x^{i_m}), y - x^{i_m} \rangle - \frac{\mu_1^{(\tau+m-1)}}{2} \|y - x^{i_m}\|_1^2 \right) \\
&\overset{\text{(i)}}{=} F(x^{i_m}) - \left(\frac{\mu_1^{(\tau+m-1)}}{2} \|\cdot\|_1^2 \right)^* (-\partial F(x^{i_m})) \\
&\overset{\text{(ii)}}{=} F(x^{i_m}) - \frac{1}{2\mu_1^{(\tau+m-1)}} \|\partial F(x^{i_m})\|_\infty^2,
\end{aligned}$$

where (i) is from the definition of conjugate function, and (ii) is from the fact that $\left(\frac{1}{2}\|\cdot\|_1^2\right)^* = \frac{1}{2}\|\cdot\|_\infty^2$ (Boyd and Vandenberghe, 2004).

More specifically,

$$F(x^*) \geq F(x^{i_m}) - \frac{1}{2\mu_1^{(\tau+m-1)}} \|\nabla f(x^{i_m}) + u\|_\infty^2 \quad \forall u \in \partial g(x^{i_m}).$$

By the definition of $Q_i(\cdot)$ in the GS-s rule (selection rule 2), we further have

$$F(x^*) \geq F(x^{i_m}) - \frac{1}{2\mu_1^{(\tau+m-1)}} \max_{i \in [d]} Q_i(x^{i_m})^2. \tag{26}$$

Recall Lemma 12, we have

$$F(x^{i_m+1}) \geq F(x^{i_m}) - \frac{1}{2L} \max_{i \in [d]} Q_i(x^{i_m})^2.$$

Plug the above equation into (26)

$$\begin{aligned}
F(x^*) &\geq F(x^{i_m}) - \frac{L}{\mu_1^{(\tau+m-1)}} (F(x^{i_m+1}) - F(x^{i_m})) \\
\Rightarrow F(x^{i_m+1}) - F^* &\leq \left(1 - \frac{\mu_1^{(\tau+m)}}{L}\right) (F(x^{i_m}) - F^*).
\end{aligned}$$

By applying the above inequality recursively, we get

$$\begin{aligned}
F(x^t) - F^* &\leq \prod_{m=1}^{k} \left(1 - \frac{\mu_1^{(\tau+m-1)}}{L}\right) (F(0) - F^*) \\
&\leq \prod_{i=1}^{\lceil \frac{t}{2} \rceil} \left(1 - \frac{\mu_1^{(\tau+i-1)}}{L}\right) (F(0) - F^*),
\end{aligned}$$

which completes the proof.

E Proof of Theorem 8

This proof is essentially the same as Theorem 4, the difference is that, by the definition of the Δ-GS-s rule (selection rule 7), the Lemma 12 becomes

$$F(x^{t+1}) - F(x^t) \leq -\frac{\Delta}{2L} \max_{i\in[d]} Q_i(x^t)^2$$

at each good step t.

Knowing that $\text{supp}(x^t) \subset W_\Delta$, we have $|\text{supp}(x^*) \cup \text{supp}(x^t)| \leq |W_\Delta|$ $\forall t > 0$. Then we can incorporate the new Lemma into the analysis of Theorem 4 and get

$$\begin{aligned} F(x^t) - F^* &\leq \left(1 - \frac{\Delta \mu_1^{(|W_\Delta|)}}{L}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F^*) \\ &\leq \left(1 - \frac{\Delta \mu_2}{|W_\Delta| L}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F^*)\,. \end{aligned}$$

F Proof of Theorem 9

Preliminaries:

Given $\Delta > 0$, we sort $W_\Delta = \{i_1, i_2, ..., i_m\}$ by the number of iteration when they first enter the working set W_Δ i.e., i_1 is the first coordinate being selected and i_2 is the second coordinate to be included in W_Δ, etc.

We denote the t-th iterate from the Δ-GCD algorithm as x^t and the t-th iterate from the totally corrective greedy algorithm (TCGA) as $\tilde{x}^t$. $W^\sharp = \{\tilde{i}_1, \tilde{i}_2, \ldots, \tilde{i}_k\}$, its elements is also sorted by the time when they enter the working set.

A claim:

First, we show that $\forall j \leq k$, there $\exists\ \epsilon_j > 0$ such that $\forall \Delta < \epsilon_j$, the first j elements in W_Δ is the same as the first j elements in $W^\sharp$.

We prove this claim by induction, when $j = 1$, $\forall \Delta \leq 1$, Δ-GCD and the TCGA both select the coordinate $\arg\max_{i\in[d]} Q_i(0)$ at the first iteration, thus the claim is true in this base case.

Assuming that the claim is true with some $j > 0$, then for $j + 1$:

By the continuity of $Q_i(\cdot)$, we know that there $\exists\ \epsilon'$ such that $\forall \|x - \tilde{x}^j\| \leq \epsilon', \arg\max_{i\in[d]} Q_i(x) = \tilde{i}_{j+1}$.

By the uniqueness (recall that $F(\cdot)$ is strongly convex) of $\tilde{x}^j$:

$$\tilde{x}^j := \underset{\text{supp}(x)\subseteq W_j}{\arg\min}\ f(x) + g(x)$$

and the optimiality condition, we also know that there $\exists\ \delta > 0$ such that $\forall x \in \mathbb{R}^d$ satisfy $\text{supp}(x) \subseteq W_j$ and $\max_{i\in W_j} Q_i(x) \leq \delta$, we have $\|x - x^j\| \leq \epsilon'$.

Denote $Q_i(x^t)$ (recall x^t is generated from Δ-GCD) is bounded by some constant B $\forall t > 0$.

Then, by setting $\Delta \leq (\min\{\epsilon_j, \delta/B\})^2$, when i_{j+1} first enter W_Δ at some iteration t, we have

$$\arg\max_{i\in W_j} Q_i(x^t) \leq \sqrt{\Delta} \arg\max_{i\in[d]} Q_i(x^t) \leq \frac{\delta}{B} B = \delta,$$

also by the induction assumption, we know that $\text{supp}(x^t) \subseteq W_j$. Putting these two conditions together, we get $\|x^t - x^j\| \leq \epsilon'$ and thus $\arg\max_{i\in[d]} Q_i(x^t) = \tilde{i}_{j+1}$, which implies that $i_{j+1} = \tilde{i}_{j+1}$. And this complete the proof of this claim.

Back to the proof:

Following the claim, we know that there $\exists\ \epsilon_k > 0$ such that for $\forall \Delta < \epsilon_k$, the first k elements in W_Δ is just $W^\sharp$.

By the nondegeneracy assumption i.e., $\delta_i > 0\ \forall x_i^* = 0$ and continuity of $Q_i(\cdot), \nabla f(\cdot)$, we know that there $\exists\ \epsilon'' > 0$ such that $\forall \|x - x^*\| < \epsilon''$ (note that $\tilde{x}^k = x^*$), $|\nabla_i f(x) - \nabla_i f(x^*)| \le \delta_i\ \forall x_i^* = 0$ and this further implies $Q_i(x) = 0\ \forall i \notin W^\sharp$ (note that $\mathrm{supp}(x^*) \in W^\sharp$).

Again, there exist $\delta'' > 0$ such that $\forall x \in \mathbb{R}^d$ satisfy $\mathrm{supp}_{W^\sharp}(x)$ and $\max_{i \in W^\sharp} Q_i(x) \le \delta''$, we have $\|x - x^*\| \le \epsilon''$.

Thus for $\Delta \le \min\{\epsilon_k, \delta''\}$, the first k elements in W_Δ will be $W^\sharp$, and any coordinate $i \notin W^\sharp$ can not be included in W_Δ. Therefore $W_\Delta = W^\sharp$.

G Proof of Theorem 5

Given the number of iteration t, denote $\mathcal{Z}_t = \{i \in [d] \mid x_i^{t'} = 0\ \forall t' < t\}$, which is the entries of x^t that filled with 0's. and $\mathcal{V}_t = \{i \in [d] \mid |\nabla_i f(x^{t'}) - \nabla_i f(x^*)| \le \delta_i\ \forall t' \ge t\}$.

From Lemma 3 (in the main text), we know that any coordinates in $\mathcal{Z}_t \cap \mathcal{V}_t$ will always stay at 0 and thus cannot be in W, that is

$$\begin{aligned} & W \subset [d] \backslash (\mathcal{Z}_t \cap \mathcal{V}_t) \quad \forall t > 0 \\ \Rightarrow & |W| \le \min_{t \in [d]} \{d - |\mathcal{Z}_t \cap \mathcal{V}_t|\}. \end{aligned} \tag{27}$$

Recall the definition of the set of good steps until the t-th iteration $\mathbb{G}_t \subset [t]$:

$$\begin{aligned} |\mathcal{V}_t| &= \sum_{i=1}^d \mathbf{1}\{|\nabla_i f(x^{t'}) - \nabla_i f(x^*)| \le \delta_i \quad \forall t' \ge t\} \\ &\ge \sum_{i=1}^d \mathbf{1}\{\|\nabla f(x^{t'}) - \nabla f(x^*)\|_\infty \le \delta_i \quad \forall t' \ge t\} \\ &\overset{\text{(i)}}{\ge} \sum_{i=1}^d \mathbf{1}\{L_\infty \|x^{t'} - x^*\|_1 \le \delta_i \quad \forall t' \ge t\} \\ &\ge \sum_{i=1}^d \mathbf{1}\{L_\infty \sup_{t' \ge t} \|x^t - x^*\|_1 \le \delta_i\}, \end{aligned} \tag{28}$$

where (i) follows from the ∞-norm smoothness assumption.

By the definition of $\mathbb{G}_t$ in section A, we also have $|\mathcal{Z}_t| \ge d - |\mathbb{G}_t|$, and further

$$\begin{aligned} |\mathcal{Z}_t \cap \mathcal{V}_t| &= |\mathcal{Z}_t| + |\mathcal{V}_t| - |\mathcal{Z}_t \cup \mathcal{V}_t| \\ &\ge d - |\mathbb{G}_t| + |\mathcal{V}_t| - d \\ &\ge |\mathcal{V}_t| - |\mathbb{G}_t|. \end{aligned} \tag{29}$$

Plug the above result in (27), we get

$$\begin{aligned} |W| &\le \min_{t>0} \{d - |\mathcal{V}_t| + |\mathbb{G}_t|\} \\ &\le \min_{t>0} \left\{ d - \sum_{i=1}^d \mathbf{1}\{L_\infty \sup_{t' \ge t} \|x^{t'} - x^*\|_1 \le \delta_i\} + |\mathbb{G}_t| \right\} \\ &\le \min_{t \in [d]} \left\{ d - \sum_{i=1}^d \mathbf{1}\{L_\infty \sup_{t' \ge t} \|x^{t'} - x^*\|_1 \le \delta_i\} + t \right\} \\ &= \min_{t \in [d]} B_t + t, \end{aligned} \tag{30}$$

where B_t is defined as $B_t := d - p_\delta\left(L_\infty \sup_{i \ge t} \left\{\|x^i - x^*\|_1\right\}\right)$ in Theorem 5.

H Proof of Corollary 6

Similar to the proof of Theorem 5, denote $\mathcal{Z}_t = \{i \in [d] \mid x_i^{t'} = 0 \ \forall t' < t\}$, which is the entries of x^t that filled with 0's. and $\mathcal{V}_t = \{i \in [d] \mid |\nabla_i f(x^{t'}) - \nabla_i f(x^*)| \leq \delta_i \ \forall t' \geq t\}$.

From Lemma 3 (in the main text), we know that any coordinates in $\mathcal{Z}_t \cap \mathcal{V}_t$ will always stay at 0 and thus cannot be in W, that is

$$\begin{aligned} &W \subset [d] \backslash (\mathcal{Z}_t \cap \mathcal{V}_t) \quad \forall t > 0 \\ \Rightarrow &|W| \leq \min_{t \in [d]} \{d - |\mathcal{Z}_t \cap \mathcal{V}_t|\}. \end{aligned} \tag{31}$$

Recall the definition of the set of good steps until the t-th iteration $\mathbb{G}_t \subset [t]$.

$$\begin{aligned} |\mathcal{V}_t| &= \sum_{i=1}^{d} \mathbf{1}\{|\nabla_i f(x^{t'}) - \nabla_i f(x^*)| \leq \delta_i \quad \forall t' \geq t\} \\ &\geq \sum_{i=1}^{d} \mathbf{1}\{\|\nabla f(x^{t'}) - \nabla f(x^*)\|_\infty \leq \delta_i \quad \forall t' \geq t\} \\ &\overset{\text{(i)}}{\geq} \sum_{i=1}^{d} \mathbf{1}\{L_\infty \|x^{t'} - x^*\|_1 \leq \delta_i \quad \forall t' \geq t\} \\ &\overset{\text{(ii)}}{\geq} \sum_{i=1}^{d} \mathbf{1}\left\{L_\infty \sqrt{\frac{2}{\mu_1}(F(x^t) - F(x^*))} \leq \delta_i \quad \forall t' \geq t\right\} \\ &\overset{\text{(iii)}}{=} \sum_{i=1}^{d} \mathbf{1}\left\{L_\infty \sqrt{\frac{2}{\mu_1}(F(x^t) - F(x^*))} \leq \delta_i\right\} \\ &\overset{\text{(iv)}}{=} p_\delta\left(L_\infty \sqrt{\frac{2}{\mu_1}(F(x^t) - F(x^*))}\right) \\ &\overset{\text{(v)}}{\geq} p_\delta\left(L_\infty \sqrt{\frac{2}{\mu_1}\prod_{i=1}^{|\mathbb{G}_t|}\left(1 - \frac{\mu_1^{(\tau+i-1)}}{L}\right)(F(0) - F^*)}\right), \end{aligned} \tag{32}$$

where (i) follows from the ∞-norm smoothness assumption, (ii) is from μ_1 strongly convex, (iii) is true since $F(x^t)$ is a decreasing sequence, (iv) is by the definition of $p_\delta(\cdot)$, (v) directly follows from Theorem 4.

By the definition of $\mathbb{G}_t$, we also have $|\mathcal{Z}_t| \geq d - |\mathbb{G}_t|$, and further

$$\begin{aligned} |\mathcal{Z}_t \cap \mathcal{V}_t| &= |\mathcal{Z}_t| + |\mathcal{V}_t| - |\mathcal{Z}_t \cup \mathcal{V}_t| \\ &\geq d - |\mathbb{G}_t| + |\mathcal{V}_t| - d \\ &\geq |\mathcal{V}_t| - |\mathbb{G}_t|. \end{aligned} \tag{33}$$

Plug the above result in (31), we get

$$\begin{aligned} |W| &\leq \min_{t>0} \{d - |\mathcal{V}_t| + |\mathbb{G}_t|\} \\ &\leq \min_{t>0} \left\{d - \left(L_\infty \sqrt{\frac{2}{\mu_1}\prod_{i=1}^{|\mathbb{G}_t|}\left(1 - \frac{\mu_1^{(\tau+i-1)}}{L}\right)(F(0) - F^*)}\right) + |\mathbb{G}_t|\right\} \\ &\leq \min_{t \in [d]} \left\{d - \left(L_\infty \sqrt{\frac{2}{\mu_1}\prod_{i=1}^{t}\left(1 - \frac{\mu_1^{(\tau+i-1)}}{L}\right)(F(0) - F^*)}\right) + t\right\} \\ &= \min_{t \in [d]} B_t + t, \end{aligned} \tag{34}$$

where B_t is defined as $B_t := d - p_\delta\left(\sqrt{\frac{2L_\infty^2}{\mu_1}\prod_{i=0}^{t-1}\left(1 - \frac{\mu_1^{(\tau+i)}}{L}\right)(F(0) - F^*)}\right)$ in Theorem 5.