Capstone Project

Coordinate Descent for Solving Linear Programs

Alina Abdikarimova

Supervisor: Dr. Adilet Otemissov
Second Reader: Dr. Zhenisbek Assylbekov

Department of Mathematics
School of Sciences and Humanities
Nazarbayev University

April 24, 2025

Abstract

Linear programming has ubiquitous applications in fields ranging from finance to engineering. Classical algorithms, such as simplex and interior point methods, are efficient for solving linear programs of moderate dimensions but face significant challenges as the dimensions of the problem grow. This capstone project addresses such challenges by proposing a coordinate descent algorithm that optimizes an unconstrained problem with quadratic penalty terms in the objective function and an additional regularization term that penalizes infeasible solutions. We will consider both the theoretical and numerical properties of the algorithm. The algorithm will be tested on randomly generated linear programs to evaluate its computational performance.

1 Introduction

Linear programming (LP) is a mathematical modelling technique used to optimize a linear objective function, subject to a set of linear constraints:

$$\begin{aligned} &\text{minimize:} && c^T x \\ &\text{subject to:} && Ax = b, \\ & && x \geq 0. \end{aligned} \tag{1}$$

It has many practical applications in engineering, economics, and other fields. Traditional algorithms like simplex and interior point methods can handle moderate-sized LP problems effectively. However, as the dimensions of the problem grow, they face significant computational challenges. Namely, the Klee-Minty example, an n-dimensional LP problem requiring $2^n - 1$ iterations, illustrates the worst-case performance of the simplex method [6]. Similarly, interior point methods are not efficient for solving high-dimensional problems with millions of constraints and variables. Since at each iteration, they solve a large linear system, they are significantly more expensive compared to the simplex method [1]. Thus, with today's growing availability of large-scale data, the need to develop new methods that efficiently solve high-dimensional LPs has become increasingly important.

One such approach that is widely used for high-dimensional problems is the coordinate descent method. Coordinate Descent (CD) is a class of first-order iterative methods in which each iterate is obtained by minimizing the objective function with respect to a single coordinate, while keeping the remaining components constant [7]. Its history dates back to the 19th century with the introduction of the Gauss-Seidel method for solving linear systems. Early discussions of its use in optimization can be found in works by Warga (1963), Ortega and Rheinboldt (1970), Luenberger (1973), Auslender (1976), and Bertsekas and Tsitsiklis (1989) [5]. In 2012, Nesterov [2] introduced a randomized version of CD and its convergence analysis for high-dimensional problems with smooth convex objective functions. In addition, a survey by Wright [7] provides a review of the main results and applications of CD.

There are deterministic and randomized ways to choose which coordinate to update at each iteration. The deterministic approach includes selecting coordinates either cyclically or according to the component of the objective function's gradient with the largest absolute value (greedy selection). Alternatively, randomized selection is made according to some probability distribution [4]. In this work, we analyze the greedy and randomized variants, where the probability distribution in the latter is determined by the relative weights of the Lipschitz constants of the gradient components, as described in [2].

To facilitate the use of coordinate descent, we reformulate problem (1) as an unconstrained optimization problem by introducing penalty terms into the objective function. As the name suggests, penalty methods are used to penalize constraint violations directly within the objective function. They can be divided into two classes: exact and inexact. In exact penalty methods, a properly chosen penalty parameter ensures that the solution to the modified problem is also a

solution to the original one. In contrast, an inexact method converges to the true solution only as the penalty parameter tends to infinity [3]. The exact penalty formulation for LPs used in this work is described in the following section.

2 Coordinate Descent Method for Linear Programs

2.1 Reformulation of LP

To solve high-dimensional LP problems using CD, we begin by formulating the unconstrained minimization problem. We first introduce quadratic penalty terms $\max\{0, -x_j\}^2$ to the objective function to handle non-negativity constraints. Then, a regularization term $\|Ax - b\|_2^2$ is added to enforce the feasibility of solutions. They are multiplied by a penalty parameter M to control the weight of the penalty terms. So, we get the unconstrained optimization problem given as:

$$\text{minimize:} \qquad c^T x + M\|Ax - b\|_2^2 + M\sum_{j=1}^{n} \max\{0, -x_j\}^2 \tag{2}$$

The reformulated objective function is expressed as

$$f(x) = \sum_{j=1}^{n} c_j x_j + M\sum_{i=1}^{m}\left(\sum_{k=1}^{n} a_{ik}x_k - b_i\right)^2 + M\sum_{j=1}^{n}\max\{0, -x_j\}^2.$$

As the convergence of CD algorithms relies on the component-wise Lipschitz continuity of the objective function's gradient, we begin by taking the partial derivative of $f(x)$ with respect to x_j:

$$\frac{\partial f(x)}{\partial x_j} = c_j + 2M\sum_{i=1}^{m} a_{ij}\left(\sum_{k=1}^{n} a_{ik}x_k - b_i\right) - 2M\max\{0, -x_j\}.$$

For the coordinate x_j, the difference in the gradients between two arbitrary points x_j and $\tilde{x}_j$ is bounded by

$$\begin{aligned}|\nabla_j f(x_j) - \nabla_j f(\tilde{x_j})| &= \left|2M\sum_{i=1}^{m} a_{ij}^2(x_j - \tilde{x}_j) - 2M\left(\max\{0, -x_j\} - \max\{0, -\tilde{x}_j\}\right)\right| \\ &\leq 2M\left(\sum_{i=1}^{m} a_{ij}^2 + 1\right)|x_j - \tilde{x}_j|.\end{aligned}$$

These bounds confirm that the gradient satisfies coordinate-wise Lipschitz continuity with $L = 2M\left(\sum_{i=1}^{m} a_{ij}^2 + 1\right)$.

Next, we describe two different strategies for choosing a coordinate at each iteration and present their convergence analysis based on Nesterov's results [2].

2.2 Convergence Analysis of Greedy Coordinate Descent

Greedy Coordinate Descent (GCD) is one of the well-known approaches for coordinate selection. It chooses the direction that leads to the greatest decrease in the objective function or one that corresponds to the gradient component with the largest magnitude. In this work, we implement the simplest variant known as the Gauss-Southwell selection rule, which chooses the coordinate with the largest absolute gradient entry [4].

Algorithm 1 Greedy Coordinate Descent

Input: $\varepsilon > 0$.

Initialization: Select an arbitrary starting point $x_0 \in \mathbb{R}^n$.

General step: For $k = 0, 1, 2, \ldots$:

1. Compute the gradient $\nabla f(x_k)$ and select the coordinate
$$j_k = \arg\max_{1 \le j \le n} |\nabla_j f(x_k)|.$$
2. Update the iterate by
$$x_{k+1} = x_k - \frac{1}{L_{j_k}} \nabla_{j_k} f(x_k) e_{j_k}.$$
3. Terminate if the infinity norm of the gradient satisfies $\|\nabla f(x_{k+1})\|_\infty \le \varepsilon$.

Given the coordinate-wise Lipschitz condition, we obtain a lower bound on the function decrease:
$$f(x_k) - f(x_{k+1}) \ge \frac{1}{2L_{j_k}} |\nabla_{i_k} f(x_k)|^2.$$
Since we choose j_k as the coordinate with the largest absolute gradient component:
$$|\nabla_{j_k} f(x_k)|^2 = \max_{1 \le j \le n} |\nabla_j f(x_k)|^2 \ge \frac{1}{n} \|\nabla f(x_k)\|^2.$$
Substituting this, we obtain:
$$f(x_k) - f(x_{k+1}) \ge \frac{1}{2nL_{j_k}} \|\nabla f(x_k)\|^2.$$
Applying the convexity inequality:
$$f(x) - f^* \le \nabla f(x)^T (x - x^*),$$
using Cauchy-Schwarz inequality and squaring both sides, we obtain:
$$\|\nabla f(x_k)\|^2 \ge \frac{(f(x_k) - f^*)^2}{\|x_k - x^*\|^2}.$$
Substituting $R \ge \|x_k - x^*\|$ gives:
$$\|\nabla f(x_k)\|^2 \ge \frac{2}{R^2} (f(x_k) - f^*)^2.$$
Substituting this, we get:
$$f(x_k) - f(x_{k+1}) \ge \frac{1}{nL_{j_k} R^2} (f(x_k) - f^*)^2.$$
$$\frac{1}{f(x_{k+1}) - f^*} - \frac{1}{f(x_k) - f^*} = \frac{f(x_k) - f(x_{k+1})}{(f(x_{k+1}) - f^*)(f(x_k) - f^*)} \ge \frac{f(x_k) - f(x_{k+1})}{(f(x_k) - f^*)^2} \ge \frac{1}{2nL_{j_k} R^2}$$
$$\frac{1}{f(x_k) - f^*} \ge \frac{1}{f(x_{k-1}) - f^*} + \frac{1}{2nL_{j_{k-1}} R^2} \ge \frac{1}{f(x_{k-2}) - f^*} + \frac{1}{2nL_{j_{k-1}} R^2} + \frac{1}{2nL_{j_{k-2}} R^2} \ge$$
$$\ge \ldots \ge \frac{1}{f(x_0) - f^*} + \frac{k}{2nL_{max} R^2}$$

Using (7):

$$f(x_0) \geq f^* + \langle \nabla f(x^*), x_0 - x^* \rangle + \frac{1}{2} S_\alpha \|x_0 - x^*\|_{1-\alpha}^2 = f^* + \frac{1}{2} \sum_{j=1}^{n} L_j^\alpha (L_j^{1-\alpha} \|(x_0 - x^*)^{(j)}\|_{(j)}^2 =$$

$$= f^* + \frac{1}{2} \sum_{j=1}^{n} L_j \|(x_0 - x^*)^{(j)}\|_{(j)}^2 \leq f^* + \frac{1}{2} n L_{max} R^2$$

Therefore,

$$f(x_k) - f^* \leq \frac{2nL_{max}R^2}{k+4}, \quad k \geq 0. \tag{3}$$

It indicates that the convergence is sublinear at a rate of $O(1/k)$ and will be compared with the actual convergence in the next section. Also, since the bound is inversely proportional to the number of iterations k, the method converges in general, with the rate depending on problem-specific constants.

2.3 Convergence Analysis of Random Coordinate Descent

Another common strategy is Random Coordinate Descent (RCD), where each coordinate is proportionally weighed according to its component-wise Lipschitz gradient constants.

For the analysis of RCD, we consider (2) under the assumption that its optimal set is nonempty and bounded. Fix a decomposition of $\mathbb{R}^N$ on n subspaces:

$$\mathbb{R}^N = \bigotimes_{i=j}^{n} \mathbb{R}^{n_j}, \quad N = \sum_{i=1}^{n} n_j.$$

Consider the partition of the identity matrix:

$$I_N = (U_1, \ldots, U_n) \in \mathbb{R}^{N \times N}, \quad U_j \in \mathbb{R}^{N \times n_j}, \quad j = 1, \ldots, n.$$

Hence, any vector $x = (x^{(1)}, \ldots, x^{(n)})^T \in \mathbb{R}^N$ can be expressed as:

$$x = \sum_{j=1}^{n} U_j x^{(j)}, \quad x^{(j)} \in \mathbb{R}^{n_j}, \quad j = 1, \ldots, n.$$

Consequently, the partial gradient of $f(x)$ with respect to $x^{(j)}$ is:

$$f_j'(x) = U_j^T \nabla f(x) \in \mathbb{R}^{n_j}, \quad x \in \mathbb{R}^N.$$

For $\mathbb{R}^{n_j}$, we fix some norms $\|\cdot\|_{(j)}, j = 1, \ldots, n$. Given this setup, the gradient of objective function f in (2) is coordinate-wise Lipschitz continuous with corresponding constants L_j:

$$\|f_j'(x + U_j h_j) - f_j'(x)\|_{(j)}^* \leq L_j \|h_j\|_{(j)}, \quad h_j \in \mathbb{R}^{n_j}, \quad j = 1, \ldots, n, \quad x \in \mathbb{R}^N.$$

where $L_j = 2M \left(\sum_{i=1}^{m} a_{ij}^2 + 1 \right)$. Using this, we bound:

$$\langle f_j'(x + tU_j h_j), h_j \rangle \leq \langle f_j'(x), h_j \rangle + L_j t \|h_j\|_{(j)}^2.$$

Differentiating $f(x + tU_j h_j)$:

$$\frac{d}{dt} f(x + tU_j h_j) = \langle f_j'(x + tU_j h_j), h_j \rangle.$$

Integrating from $t = 0$ to $t = 1$:

$$f(x + U_j h_j) - f(x) \leq \int_0^1 (\langle f_j'(x), h_j \rangle + L_j t \|h_j\|_{(j)}^2) dt.$$

Evaluating the integral:

$$f(x + U_j h_j) - f(x) \leq \langle f_j'(x), h_j \rangle + \frac{L_j}{2} \|h_j\|_{(j)}^2.$$

$$f(x + U_j h_j) \leq f(x) + \langle f_j'(x), h_j \rangle + \frac{L_j}{2} \|h_j\|_{(j)}^2, \quad x \in \mathbb{R}^N, \quad h_j \in \mathbb{R}^{n_j}, \quad j = 1, \ldots, n. \tag{4}$$

Algorithm 2 Random Coordinate Descent

Input: $\varepsilon > 0$.
Initialization: Select an arbitrary starting point $x_0 \in \mathbb{R}^n$.
General step: For $k = 0, 1, 2, \ldots$:

1. Choose a coordinate index:
$$j_k = \mathcal{R}_\alpha.$$
2. Update the iterate using the operator T_{j_k}:
$$x_{k+1} = T_{j_k}(x_k).$$
3. Terminate if the infinity norm of the gradient satisfies $\|\nabla f(x_{k+1})\|_\infty \leq \varepsilon$

Here, the optimal coordinate steps are defined in the following way:

$$T_j(x) \stackrel{\text{def}}{=} x - \frac{1}{L_j} U_j f_j'(x)^\#, \quad j = 1, \ldots, n,$$

where the notation $s^\#$ refers to an arbitrary vector that is chosen from the set:

$$s^\# \in \arg\max_x \left[\langle s, x \rangle - \frac{1}{2} \|x\|^2 \right].$$

It attains the dual norm of s, that is,

$$\|s^\#\| = \|s\|^* = \max_{\|h\|=1} \langle s, x \rangle.$$

The dual of the Euclidean norm is the Euclidean norm itself, meaning that for any vector s, we have:

$$\|s\|^* = \|s\|.$$

To determine $s^\#$, we solve the maximization problem:

$$s^\# \in \arg\max_x \left\{ s^T x - \frac{1}{2} x^T x \right\}.$$

Taking the gradient with respect to x and setting it to zero:

$$s - x = 0 \Rightarrow x = s.$$

Thus, we get:

$$s^\# = s.$$

Therefore, the optimal coordinate step simplifies to:

$$T_j(x) = x - \frac{1}{L_j} U_j f_j'(x).$$

Using the bound (4), we obtain

$$f(x) - f(T_j(x)) \geq \frac{1}{2L_j} \left(\|f_j'(x)\|_{(j)}^* \right)^2, \quad j = 1, \ldots, n. \tag{5}$$

Next, we proceed by defining a random counter, $\mathcal{R}_\alpha, \alpha \in \mathbb{R}$, that selects an integer $j \in \{1, \ldots, n\}$ with probability

$$p_j^{(\alpha)} = L_j^\alpha \left[\sum_{j=1}^n L_j^\alpha \right]^{-1}, \quad j = 1, \ldots, n.$$

The lemma below establishes an important inequality for analyzing the convergence of RCD.

Lemma 1 *Let the gradient of a function f be coordinate-wise Lipschitz continuous with constants $L_j = L_j(f)$. Then for any $\alpha \in \mathbb{R}$ we have*

$$\|\nabla f(x) - \nabla f(y)\|_{1-\alpha}^* \leq S_\alpha \|x - y\|_{1-\alpha}, \quad x, y \in \mathbb{R}^N. \tag{6}$$

Therefore,

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2} S_\alpha \|x - y\|_{1-\alpha}^2, \quad x, y \in \mathbb{R}^N. \tag{7}$$

Proof:

$$f(x) - f^* \overset{(5)}{\geq} \max_{1 \leq j \leq n} \frac{1}{2L_j} \left(\|f_j'(x)\|_{(j)}^* \right)^2 \geq \frac{1}{2S_\alpha} \sum_{j=1}^n \frac{L_j^\alpha}{L_j} \left(\|f_j'(x)\|_{(j)}^* \right)^2 = \frac{1}{2S_\alpha} \left(\|\nabla f(x)\|_{1-\alpha}^* \right)^2.$$

Applying this inequality to $f(x) - f(y) - \langle \nabla f(y), x - y \rangle$, we get

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2S_\alpha} \left(\|\nabla f(x) - \nabla f(y)\|_{1-\alpha}^* \right)^2, \quad x, y \in \mathbb{R}^N. \tag{8}$$

Adding two variants of (8) with x and y interchanged, and using the Cauchy-Schwarz inequality, we obtain

$$\frac{1}{S_\alpha} \left(\|\nabla f(x) - \nabla f(y)\|_{1-\alpha}^* \right)^2 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle \leq \|\nabla f(x) - \nabla f(y)\|_{1-\alpha}^* \cdot \|x - y\|_{1-\alpha}.$$

Rearranging:

$$\|\nabla f(x) - \nabla f(y)\|_{1-\alpha}^* \leq S_\alpha \|x - y\|_{1-\alpha}$$

To derive inequality (6) from (5), we define:

$$\psi(t) = f(y + t(x - y)), \quad t \in [0, 1].$$

Applying the fundamental theorem of calculus and the chain rule:

$$f(x) - f(y) = \int_0^1 \langle \nabla f(y + t(x - y)), x - y \rangle dt.$$

Splitting the integral:

$$f(x) - f(y) = \langle \nabla f(y), x - y \rangle + \int_0^1 \langle \nabla f(y + t(x - y)) - \nabla f(y), x - y \rangle dt.$$

From (5),

$$\|\nabla f(y + t(x - y)) - \nabla f(y)\|^*_{1-\alpha} \leq S_\alpha t \|x - y\|_{1-\alpha}.$$

Applying Cauchy-Schwarz inequality and integrating:

$$\int_0^1 \langle \nabla f(y + t(x - y)) - \nabla f(y), x - y \rangle dt \leq \frac{1}{2} S_\alpha \|x - y\|^2_{1-\alpha}.$$

Thus,

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2} S_\alpha \|x - y\|^2_{1-\alpha}.$$

□

After k iterations, *RCD* produces a random output $(x_k, f(x_k))$, which depends on the particular sequence of coordinates selected during the iterations

$$\xi_k = \{j_0, \ldots, j_k\}.$$

We will show that the expected value

$$\phi_k \stackrel{\text{def}}{=} \mathbb{E}_{\xi_{k-1}} f(x_k)$$

converges to the optimal solution of the problem (2).

Theorem 1 *For any $k \geq 0$ we have*

$$\phi_k - f^* \leq \frac{2}{k+4} S_\alpha R^2_{1-\alpha}(x_0), \tag{2.12}$$

where

$$R_\beta(x_0) = \max_x \left\{ \max_{x_* \in X^*} \|x - x_*\|_\beta : f(x) \leq f(x_0) \right\}.$$

Proof:

Let *RCD* generate the random variable x_k at iteration k. Then

$$f(x_k) - E_{j_k}(f(x_{k+1})) = \sum_{j=1}^n p_\alpha^{(j)} \cdot [f(x_k) - f(T_j(x_k))] \stackrel{(5)}{\geq} \sum_{j=1}^n \frac{p_\alpha^{(j)}}{2L_j} \left(\|f_j'(x_k)\|^*_{(j)} \right)^2 = \frac{1}{2S_\alpha} (\|\nabla f(x_k)\|^*_{1-\alpha})^2.$$

Noting that $f(x_k) \leq f(x_0)$ and using the Cauchy-Schwarz inequality:

$$f(x_k) - f^* \leq \min_{x_* \in X^*} \langle \nabla f(x_k), x_k - x_* \rangle \leq \|\nabla f(x_k)\|^*_{1-\alpha} R_{1-\alpha}(x_0).$$

Hence,

$$f(x_k) - \mathbb{E}_{j_k}(f(x_{k+1})) \geq \frac{1}{2S_\alpha R^2_{1-\alpha}(x_0)} (f(x_k) - f^*)^2,$$

Taking expectation over ξ_{k-1} and applying Jensen's inequality, we obtain

$$\phi_k - \phi_{k+1} \geq \frac{1}{2S_\alpha R^2_{1-\alpha}(x_0)} \mathbb{E}_{\xi_k}[(f(x_k) - f^*)^2] \geq \frac{1}{2S_\alpha R^2_{1-\alpha}(x_0)} (\phi_k - f^*)^2.$$

Therefore,

$$\frac{1}{\phi_{k+1} - f^*} - \frac{1}{\phi_k - f^*} = \frac{\phi_k - \phi_{k+1}}{(\phi_{k+1} - f^*)(\phi_k - f^*)} \geq \frac{\phi_k - \phi_{k+1}}{(\phi_k - f^*)^2} \geq \frac{1}{2S_\alpha R^2_{1-\alpha}(x_0)},$$

Finally, we get

$$\frac{1}{\phi_k - f^*} \geq \frac{1}{\phi_0 - f^*} + \frac{k}{2S_\alpha R^2_{1-\alpha}(x_0)} \stackrel{(7)}{\geq} \frac{k+4}{2S_\alpha R^2_{1-\alpha}(x_0)} \quad \text{for any } k \geq 0.$$

□

The implementation of RCD with $\alpha = 1$ is provided in the following section, and its empirical convergence will be compared with the expected rate of convergence, which is given as:

$$\phi_k - f^* \leq \frac{2}{k+4} \left[\frac{1}{n} \sum_{j=1}^{n} L_j \right] R_0^2(x_0) \tag{9}$$

This bound also indicates that the convergence is sublinear at a rate of $O(1/k)$.

3 Numerical Experiments

We conducted a series of numerical experiments on random problems to study the practical performance of GCD and RCD (with $\alpha = 1$). To construct random LPs, we generate random primal and dual feasible points that satisfy the Karush-Kuhn-Tucker conditions. A similar approach to randomly generating inequality-constrained LPs is discussed in [6]. For arbitrary positive integers m and n, we proceed as follows:

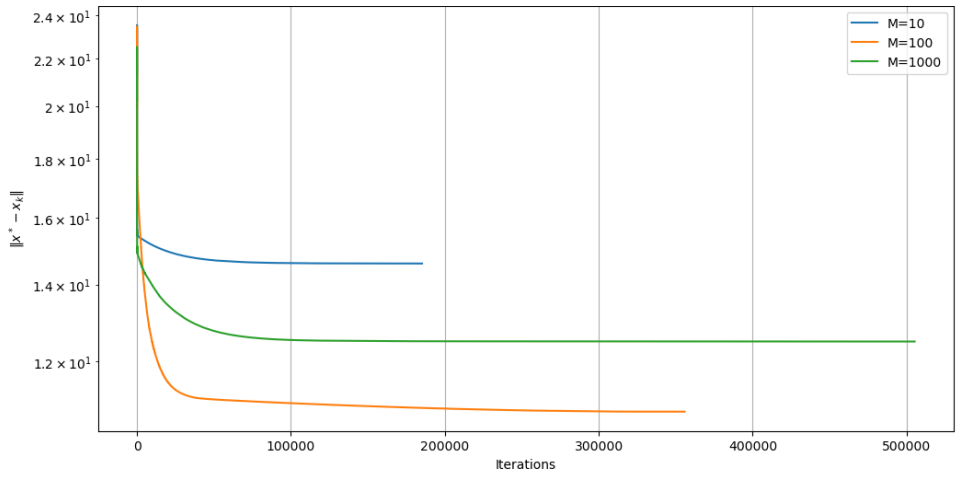
```
x_star = np.round(10 * np.random.rand(n))
y_star = np.round(10 * np.random.randn(m))
z_star = np.round(10 * np.random.rand(n))
z_star[x_star > 0] = 0
A = np.round(10 * np.random.randn(m, n))
b = A @ x_star
c = A.T @ y_star + z_star
```

It ensures that:

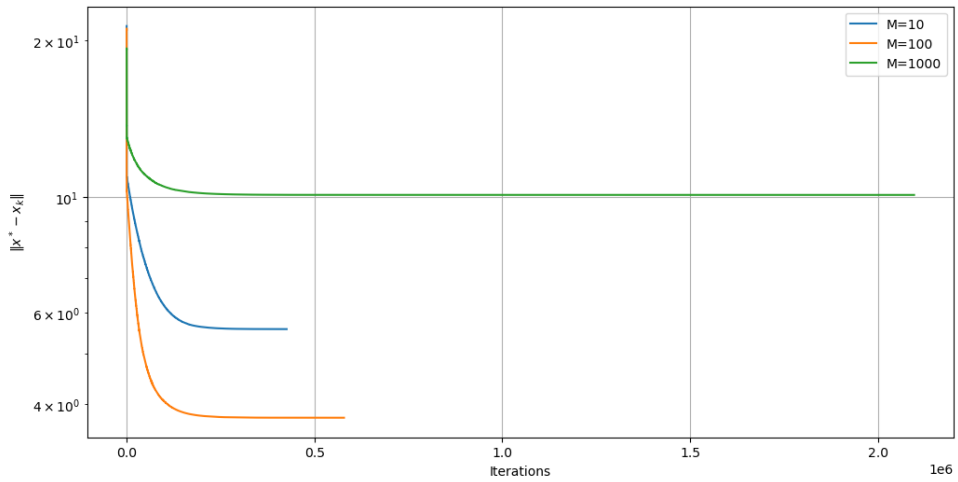
- a primal solution $x^* \in \mathbb{R}^n$ is sampled from a uniform distribution to ensure non-negativity;
- a dual variable $y^* \in \mathbb{R}^m$ is sampled from a standard normal distribution;
- a slack variable $z^* \in \mathbb{R}^n$ is also sampled from a uniform distribution and modified to satisfy the complementarity slackness condition by setting $z_j^* = 0$ whenever $x_j^* > 0$;
- the matrix $A \in \mathbb{R}^{m \times n}$ is sampled from a standard normal distribution;
- vectors b and c are defined as $b = Ax^*$ and $c = A^\top y^* + z^*$ to ensure primal and dual feasibility.

3.1 Impact of M on Convergence

We consider a fixed problem instance with $m = 10$ and $n = 15$, and study the convergence behaviour of GCD and RCD for different penalty parameter values $M \in \{10, 100, 1000\}$. For each case, we plot the norm of the error vector $\|x^* - x_k\|$ against the number of iterations for each method.



(a)



(b)

Figure 1: Convergence of $\|x^* - x_k\|$ over iterations for (a) GCD and (b) RCD.

Table 1: Performance of Coordinate Descent Methods for Different Penalty Parameters

Method	M	**Objective Value**	**Time (s)**	**Iterations**
GCD	10	-2031	29.659	185,189
	100	-1969.1	52.8977	355,746
	1000	-1963.68	95.4447	505,238
RCD	10	-2031	97.1183	425,966
	100	-1969.8	153.59	578,565
	1000	-1963.68	451.399	2,096,177

Note: The true objective value is -1963.0.

From Figure 1 and Table 1, we observe that increasing M leads to an increase in the number of iterations required but enhances the accuracy of the objective function value. However, this improvement does not always extend to the solution vector x.

3.2 Impact of Dimension

To evaluate the scalability of GCD and RCD, we fix the penalty parameter $M = 100$. First, the number of variables is held constant at $n = 25$, while the number of variables m is increased from 5 to 20. Second, we scale both m and n while keeping $m/n = 5/7$. In both experiments, 5 random problems are generated for each (m, n), and the results are averaged to reduce the impact of randomness.

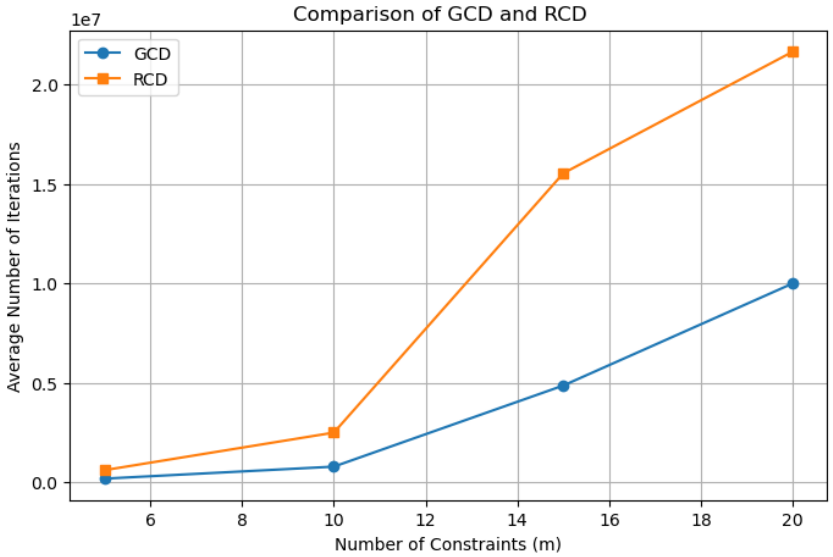


Figure 2: Number of Iterations vs. Number of Constraints for GCD and RCD

As shown in Figure 2, while both algorithms require more iterations as the number of constraints m increases, GCD converges faster than RCD.

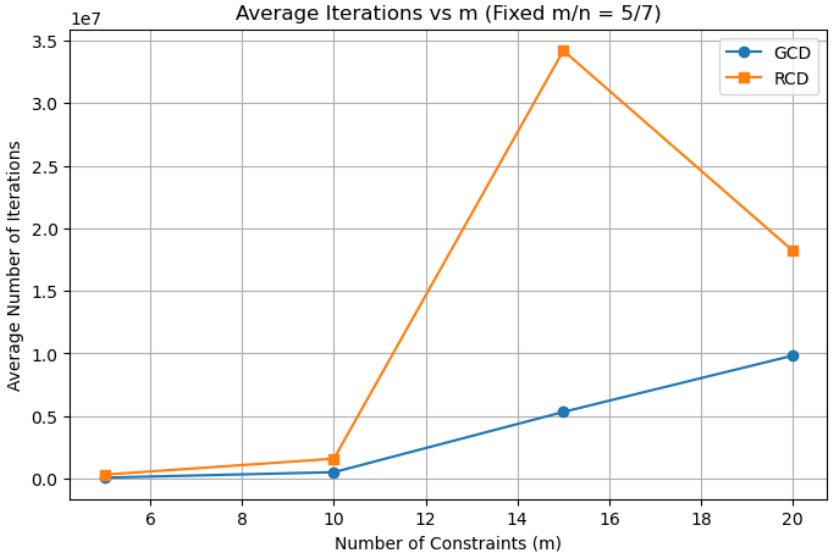


Figure 3: Number of Iterations vs. Problem Size for GCD and RCD (m/n=5/7)

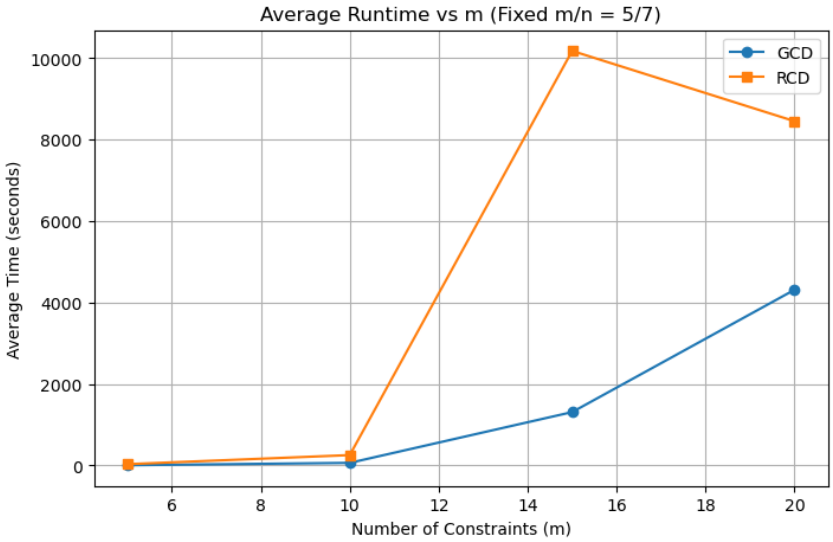


Figure 4: Execution Time vs. Problem Size for GCD and RCD (m/n=5/7)

From Figure 3 and Figure 4, it can be seen that GCD again outperforms RCD in terms of both the iteration count and the execution time. Also, as the problem size increases, both methods take more iterations to converge. However, RCD shows a sharp increase at $m = 15$, then drops at $m = 20$, possibly due to the randomness of the problems.

3.3 Comparison with Theoretical Upper Bounds

To see how GCD and RCD perform compared to theory, we compare their experimental convergence plots with the corresponding theoretical upper bounds provided in Section 2 (equations (3) and (9)). The comparison is based on a fixed problem instance with $m = 10$, $n = 15$, and $M = 100$.

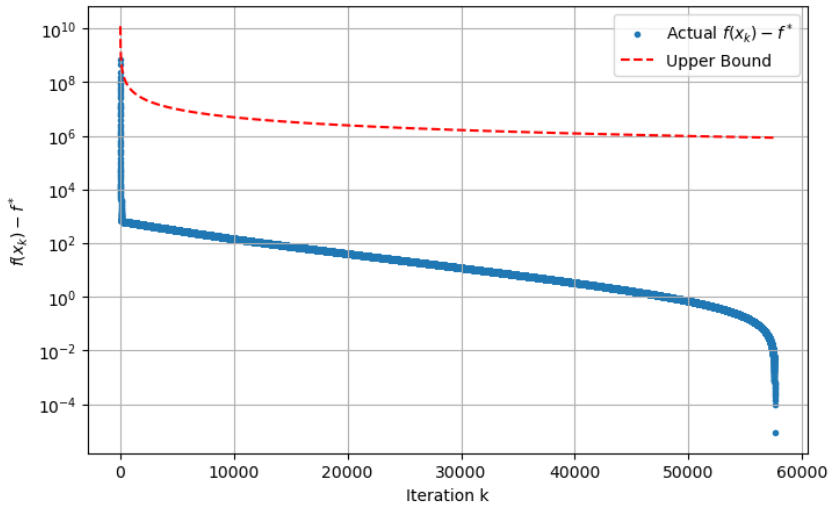


Figure 5: Experimental vs. Theoretical Convergence of GCD

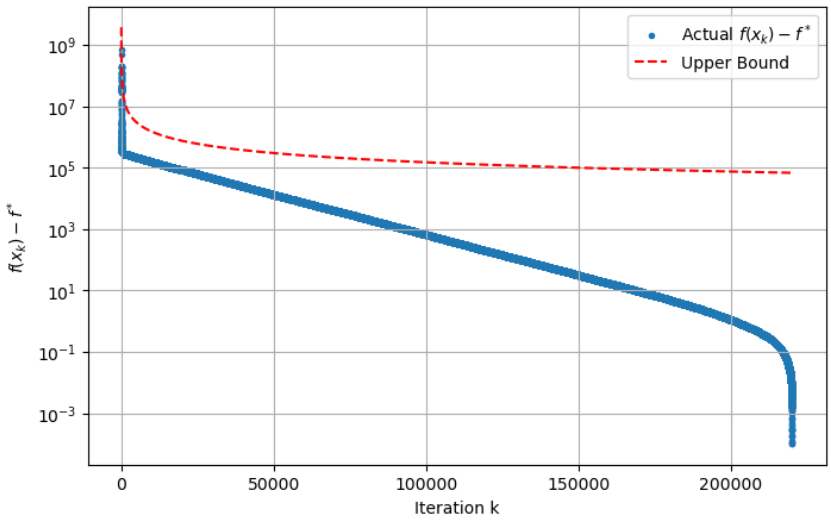


Figure 6: Experimental vs. Theoretical Convergence of RCD

The convergence rate for both methods is much faster than the provided upper bounds. Their convergence behaviour can be divided into three phases. Initially, there is a rapid improvement in the objective value. GCD picks the coordinate with the largest gradient, the one that will give the biggest improvement. RCD is also likely to pick useful coordinates due to randomness, especially with $\alpha = 1$, where the selection is biased toward coordinates with larger Lipschitz constants. In the middle phase, improvements slow down as the penalty terms become harder to reduce. Finally, the algorithms rapidly converge once they approach the optimal solution.

4 Conclusion

This project investigated the application of coordinate descent methods for solving linear programs reformulated as unconstrained optimization problems using penalty terms. We analyzed the convergence of both greedy and randomized versions and tested them on randomly generated problems. Although our approach demonstrates significantly slower convergence compared to classical LP solvers, it provided a starting point for future work exploring alternative formulations. In particular, it led to the development of an alternative unconstrained reformulation with improved convergence that avoids the penalty parameter M:

$$\begin{aligned}\arg\min_{x,\lambda,s} f(x,\lambda,s) :=& \frac{1}{2}(c^T x - b^T\lambda)^2 + \frac{1}{2}||Ax-b||_2^2 + \frac{1}{2}||A^T\lambda + s - c||_2^2 \\ &+ \sum_{j=1}^{n} \max\{-x_j, 0\}^q + \max\{-s_j, 0\}^q,\end{aligned}$$

where $q > 2$ ensures that the objective function is twice continuously differentiable.

References

[1] Jacek Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

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

[3] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2 edition, 2006.

[4] Hao-Jun Shi, Shenyinying Tu, Yangyang Xu, and Wotao Yin. A primer on coordinate descent algorithms. 2016.

[5] Ryan J. Tibshirani. Dykstra's algorithm, admm, and coordinate descent: connections, insights, and extensions. page 517–528, 2017.

[6] Robert J. Vanderbei. *Linear Programming: Foundations and Extensions*. International Series in Operations Research & Management Science. Springer Cham, 5 edition, 2020.

[7] Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.