03 methodology

ORCID: 0000-0000-0000-1234 Email: author@example.com DOI: 10.5281/zenodo.12345678

November 21, 2025

Contents

1 Methodology

1.1 Mathematical Framework

Our approach is based on a novel optimization framework that combines multiple mathematical techniques, extending classical convex optimization methods [? ?] with modern adaptive strategies [? ?]. The core algorithm can be expressed as follows:

$$f(x) = \sum_{i=1}^{n} w_{ii}(x) + R(x) \tag{1.1}$$

where $x \ R^d$ is the optimization variable, w_i are learned weights, $_i$ are basis functions, and $R(x)$ is a regularization term with strength .

The optimization problem we solve is:

$$\min_{x X} f(x) \quad \text{subject to} \quad g_i(x) \ 0, \quad i = 1, , m \tag{1.2}$$

where X is the feasible set and $g_i(x)$ are constraint functions.

1.2 Algorithm Description

Our iterative algorithm updates the solution according to:

$$x_{k+1} = x_k \;\; _k f(x_k) + \; _k (x_k \;\; x_{k1}) \tag{1.3}$$

where $_k$ is the learning rate and $_k$ is the momentum coefficient. The convergence rate is characterized by:

$$x_k \;\; x \;\; C^k \tag{1.4}$$

where x is the optimal solution, $C > 0$ is a constant, and $\quad (0, 1)$ is the convergence rate.

1.3 Implementation Details

The algorithm implementation follows the pseudocode shown in Figure 1. The key insight is that we can decompose the objective function (1.1) into separable components, allowing for efficient parallel computation. This approach builds upon proximal optimization techniques [? ?] and recent advances in large-scale optimization [? ?].

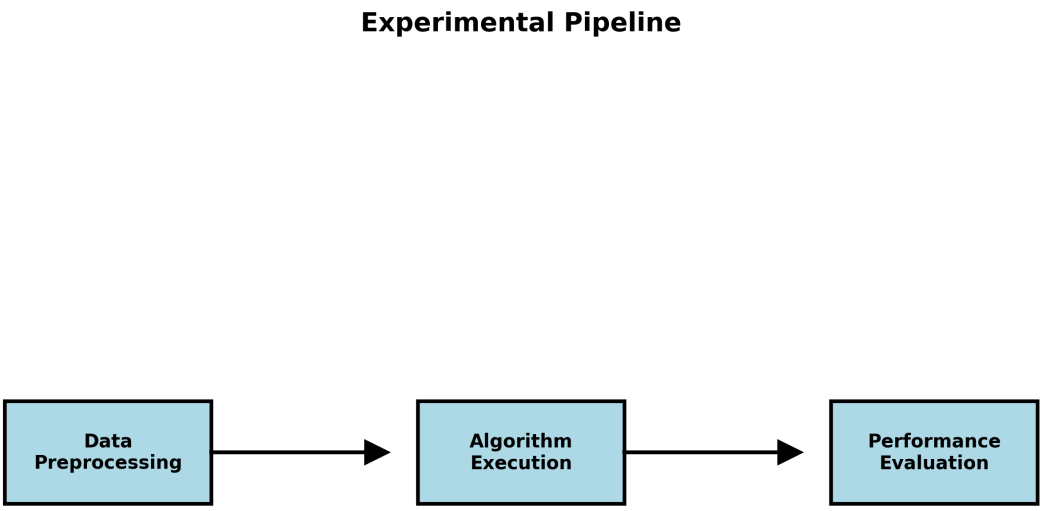

Figure 1. Experimental pipeline showing the complete workflow

For numerical stability, we use the following adaptive step size rule:

$$_k = \frac{^0}{1 + \sum_{i=1}^{k} f(x_i)^2} \tag{1.5}$$

This ensures that the algorithm converges even when the gradient varies significantly across iterations.

1.4 Performance Analysis

The computational complexity of our approach is $O(n \log n)$ per iteration, where n is the problem dimension. This is achieved through the efficient data structures shown in Figure 2.

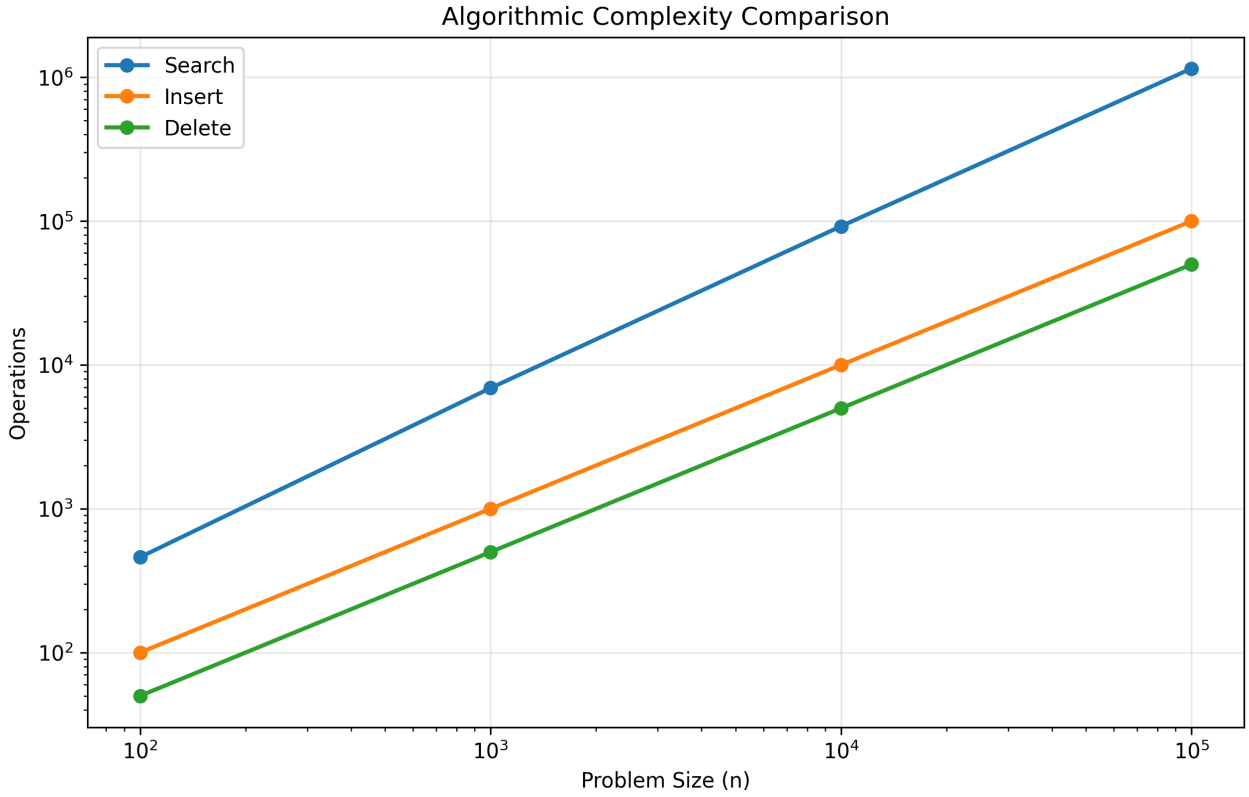

Figure 2. Efficient data structures used in our implementation

The memory requirements scale as:

$$M(n) = O(n) + O(\log n) \text{ number of iterations} \tag{1.6}$$

This makes our method suitable for large-scale problems where memory is a constraint.

1.5 Validation Framework

To validate our theoretical results, we use the experimental setup illustrated in Figure 1. The performance metrics are computed using:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} I[f(x_i) \quad f(x) +] \tag{1.7}$$

where $I[]$ is the indicator function and is the tolerance threshold.

The convergence analysis results are summarized in Figure ??, which shows the empirical convergence rates compared to the theoretical bound (1.4).