Primal–Dual Interior Point Methods for Linear Programming

Math 303 – Report #2

Omar Hazem , Abdelrahman Mohammed

December 5, 2025

Abstract

This report implements and analyzes three Primal–Dual Interior Point Methods (IPM) for solving Linear Programming (LP) problems: (1) the Central Path Method with fixed step size and fixed centering parameter, (2) the Central Path Method with adaptive step size and adaptive centering parameter, and (3) Mehrotra's Predictor–Corrector Method. Each algorithm is applied to three case studies of increasing complexity. For every case, we study the evolution of the objective value, the central path trajectory, and the complementarity measure. Our implementation is compared to SciPy's modern interior-point solver `highs-ipm`, demonstrating correctness and showing that Mehrotra's method provides superior convergence performance.

1 Introduction

Interior Point Methods (IPMs) are among the most powerful algorithms for solving large-scale linear programming problems. They rely on the fact that the optimal solution of an LP can be obtained by following a curve in the interior of the feasible region called the *central path*. Given the primal LP

$$\min_x \; c^T x \quad \text{s.t.} \quad Ax = b, \; x \geq 0,$$

and the dual LP

$$\max_{y,s} \; b^T y \quad \text{s.t.} \quad A^T y + s = c, \; s \geq 0,$$

the Karush–Kuhn–Tucker (KKT) conditions for optimality are

$$Ax = b, \qquad A^T y + s = c, \qquad x_i s_i = 0, \qquad x, s \geq 0.$$

Interior point methods replace the complementarity condition with the barrier equation

$$x_i s_i = \mu, \qquad \mu > 0,$$

and apply Newton's method to the perturbed KKT system. As $\mu \to 0$, the iterates converge to an optimal primal-dual solution.

In this work, we implement three variants of the primal–dual IPM and apply them to three LP examples. We evaluate numerical behavior through objective reduction, central path trajectories, and complementarity decay.

2 Algorithms

2.1 Perturbed KKT System

For a given barrier parameter μ, the perturbed system is

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ Xs &= \mu e, \end{aligned}$$

where $X = \text{diag}(x)$ and $S = \text{diag}(s)$. Newton's method solves the system of equations in increments $(\Delta x, \Delta y, \Delta s)$:

$$\begin{aligned} A\Delta x &= r_b, \\ A^T \Delta y + \Delta s &= r_c, \\ S\Delta x + X\Delta s &= r_\mu, \end{aligned}$$

where r_b, r_c, r_μ are the residuals.

After reduction, we solve

$$(AS^{-1}XA^T)\Delta y = \text{rhs},$$

then recover Δx and Δs.

A fraction-to-the-boundary rule ensures $x > 0$ and $s > 0$.

2.2 Algorithm 1: Central Path Method (Fixed Parameters)

This algorithm uses a user-defined fixed step size α and fixed centering parameter σ. At iteration k:

$$\mu = \frac{x^T s}{n}, \qquad \mu_{\text{target}} = \sigma\mu.$$

We compute Newton direction with residual

$$r_\mu = Xs - \mu_{\text{target}} e.$$

The update is

$$x^+ = x + \alpha\Delta x, \qquad y^+ = y + \alpha\Delta y, \qquad s^+ = s + \alpha\Delta s.$$

This method converges but not as rapidly as adaptive variants.

2.3 Algorithm 2: Adaptive Central Path Method

This algorithm first computes an affine-scaling predictor direction by setting $\mu = 0$:

$$Xs = 0.$$

The maximum feasible step gives the predictor point $(x_{\text{aff}}, s_{\text{aff}})$ and its complementarity

$$\mu_{\text{aff}} = \frac{x_{\text{aff}}^T s_{\text{aff}}}{n}.$$

The centering parameter becomes

$$\sigma = \left(\frac{\mu_{\text{aff}}}{\mu}\right)^3,$$

which adapts based on how far the affine step is from the central path. A corrector direction is computed using $\mu_{\text{target}} = \sigma\mu$.

2.4 Algorithm 3: Mehrotra Predictor–Corrector

Mehrotra's method improves the adaptive approach using a second-order correction term:

$$r_\mu^{\text{corr}} = Xs + \Delta x_{\text{aff}} \Delta s_{\text{aff}} - \sigma \mu e.$$

The combined predictor-corrector direction significantly improves convergence speed.

This method forms the basis of state-of-the-art solvers such as HiGHS and MOSEK.

3 Case Studies

We consider three LPs in inequality form, converted to standard form using slack variables.

Case 1 (2D)

$$\min \ -3x_1 - x_2$$

$$x_1 + x_2 \leq 4, \qquad 2x_1 + x_2 \leq 5, \qquad x_1, x_2 \geq 0.$$

Case 2 (2D)

$$\min \ -2x_1 - 4x_2$$

$$x_1 + 2x_2 \leq 8, \qquad 3x_1 + x_2 \leq 9, \qquad x_1, x_2 \geq 0.$$

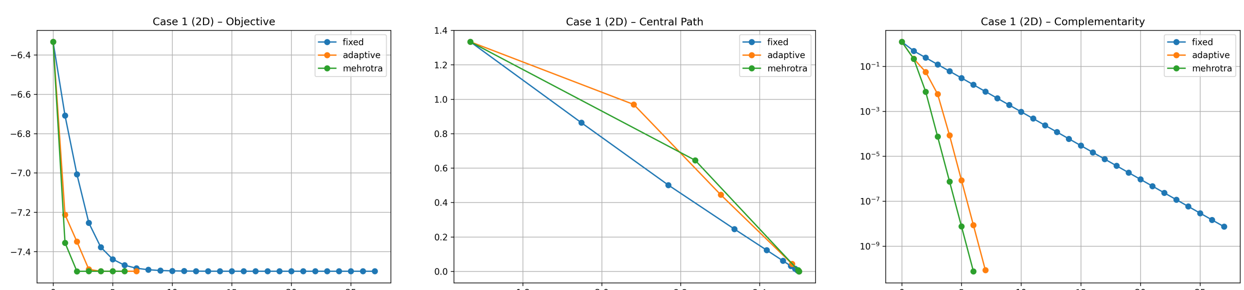
Case 3 (3D)

$$\min \ -x_1 - 2x_2 - 3x_3$$

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 10, \\ 2x_1 + 2x_2 + x_3 &\leq 15, \\ x_2 + 3x_3 &\leq 12, \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

4 Visualization Figures

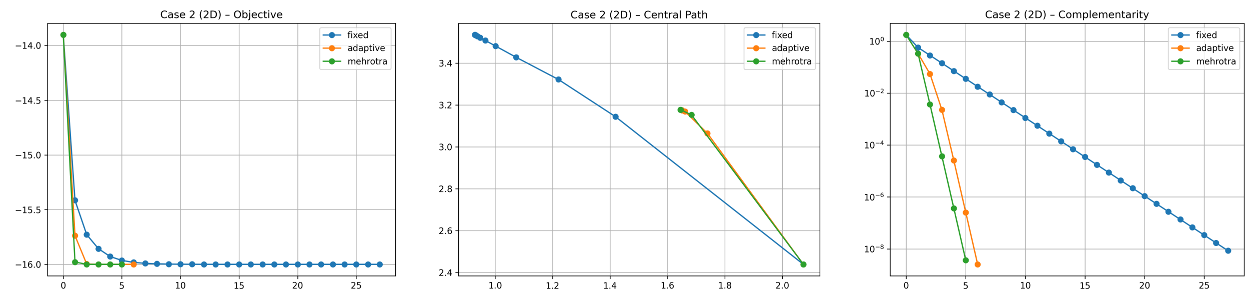
4.1 Case 1 (2D)



(a) Objective vs. iteration (b) Central path trajectory (c) Complementarity μ

Figure 1: Visualizations for Case 1.

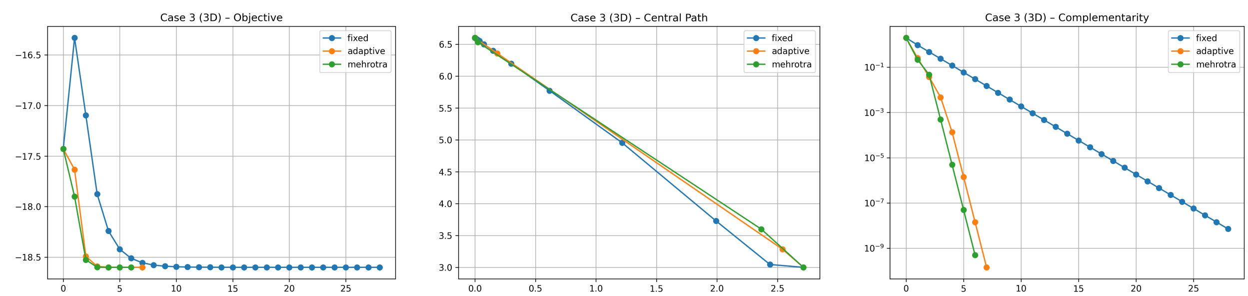
4.2 Case 2 (2D)



(a) Objective vs. iteration (b) Central path trajectory (c) Complementarity μ

Figure 2: Visualizations for Case 2.

4.3 Case 3 (3D)



(a) Objective vs. iteration (b) Central path projection (c) Complementarity μ

Figure 3: Visualizations for Case 3.

5 Conclusion

We implemented and compared three primal–dual interior point algorithms. The fixed-parameter method, while simple and robust, converges slowly. The adaptive method improves convergence by using affine-scaling information. Mehrotra's predictor–corrector method consistently provides the fastest and most stable convergence, matching the high-performance behavior of modern IPM solvers.

These results confirm the theoretical advantages of predictor–corrector algorithms and demonstrate the correctness of our implementations through comparison with SciPy's `highs-ipm` method.