Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 11(2): 4935–4965.
DOI: 10.3934/math.2026202
Received: 09 January 2026
Revised: 06 February 2026
Accepted: 11 February 2026
Published: 27 February 2026

Research article

A generalized self-regular Kernel function for large-scale nonlinear optimization problems

Mounia Laouar[1], **Mahmoud Brahimi**[1], **Raouf Ziadi**[2], **Mohammed A. Saleh**[3], **Abdulgader Z. Almaymuni**[3,*] **and Benmessaoud Chahinez**[4]

[1] Laboratory of Partial Differential Equations, Department of Mathematics, Faculty of Mathematics and Computer Science, University of Batna 2, Batna, Algeria

[2] Laboratory of Fundamental and Numerical Mathematics (LMFN), Department of Mathematics, University Setif -1- Ferhat Abbas, Setif, Algeria

[3] Department of Cybersecurity, College of Computer, Qassim University, Saudi Arabia

[4] University of Batna 2, Batna 05000, Algeria

* **Correspondence:** Email: almaymuni@qu.edu.sa.

Abstract: This work investigated the computational efficiency of primal-dual interior-point methods for nonlinear convex optimization by refining both the underlying kernel functions and the barrier parameter update mechanisms. We introduced a unified parametric class of self-regular kernels that generalizes several established barrier families while maintaining optimal theoretical iteration complexity. To bridge the gap between theoretical convergence and practical performance, we proposed an adaptive update rule for the barrier parameter and evaluated various heuristics for its dynamic selection. Extensive numerical testing on a diverse benchmark suite demonstrated that the proposed framework significantly outperforms the Interior Point OPTimizer (IPOPT) solver while maintaining high numerical accuracy and minimal stationarity residuals. Moreover, the framework exhibited robust performance even on nonconvex problems, highlighting its practical versatility beyond the theoretical convex setting.

Keywords: optimization method; nonlinear convex optimization; interior-point methods; self-regular kernels; large-scale problems
Mathematics Subject Classification: 90C20, 90C25, 90C30, 90C51

1. Introduction

Convexity is a fundamental structural property that streamlines both theoretical analysis and algorithmic development. Crucially, in the convex setting, local and global minimizers coincide; this feature renders convex optimization central to numerous disciplines in science and engineering. Such problems arise ubiquitously in portfolio optimization, power systems operation [14], and machine learning model training [6], where problem dimensions can easily exceed $m = 10000$ constraints. Recently, alternative approaches such as metaheuristics [21] and reinforcement learning [22] have been proposed for such complex systems.

Within the landscape of solvers for this problem class, interior point methods (IPMs) have gained prominence due to their polynomial-time convergence and robustness in large-scale applications [8, 16, 25]. These methods generate search directions via Newton's method applied to the Karush-Kuhn-Tucker (KKT) system, while strictly adhering to the interior of the feasible region [12, 17]. The seminal work of Peng et al. [19] introduced the concept of self-regular kernel functions, which achieve the best-known $O(\sqrt{m})$ complexity for linear optimization. These implementations often follow the predictor-corrector logic described by Mehrotra [15]. Subsequent extensions to nonlinear programming have explored various kernel designs, including trigonometric [9], hyperbolic [5], exponential [18], and hybrid formulations [10, 13, 24], each demonstrating improved practical performance under specific problem structures. Nevertheless, the choice of the kernel function remains a decisive factor in determining both their theoretical complexity and practical stability.

Despite these advances, several critical limitations persist in the current state-of-the-art methods. First, existing kernels have largely been developed in isolation, each accompanied by its own complexity analysis. This fragmentation precludes systematic comparison and hinders the principled design of new kernels with prescribed properties. Second, state-of-the-art solvers such as Interior Point OPTimizer (IPOPT) [23], which remains the reference implementation for nonlinear IPMs, widely adopted in engineering applications and numerical libraries, continue to rely predominantly on fixed barrier reduction strategies in their default configuration. While Interior Point OPTimizer (IPOPT) builds upon trust-region interior-point techniques [7] and offers adaptive barrier updates based on dual infeasibility measures, its default static strategy limits systematic adaptation to problem-specific structure, leaving significant room for improvement through more aggressive adaptive approaches.

Beyond these practical limitations, a fundamental gap persists between the $O(\sqrt{m})$ worst-case complexity bound [20] and the markedly superior performance observed in numerical practice. While empirical studies consistently report iteration counts scaling as $O(m^{\alpha})$ with $\alpha < 0.5$, a comprehensive theoretical framework explaining this discrepancy across diverse problem dimensions remains elusive. For instance, while theory predicts approximately 100 iterations for $m = 10000$ constraints, state-of-the-art solvers typically converge in 20–30 iterations, suggesting underlying structure not captured by worst-case bounds.

To address these challenges, this work introduces a unified IPM framework based on a general parametric family of kernel functions. This family subsumes many existing kernels as special cases and enables the derivation of necessary and sufficient conditions for self-regularity [19]. When combined with adaptive barrier parameter reduction strategies, our approach maintains the $O(\sqrt{m})$ worst-case complexity bound while demonstrating speedups of up to 37× over IPOPT on large-scale test problems.

Our main contributions are summarized as follows:

(1) A unified kernel framework: We introduce a generalized parametric class of self-regular kernel functions that subsumes five major kernel families [5, 9, 10, 13, 18] as special cases. This

framework provides explicit conditions on the constituent functions (monotonicity, convexity, and growth constraints), enabling a systematic approach to kernel design with provable complexity guarantees.

(2) Adaptive barrier strategies: We propose and analyze iteration-dependent barrier reduction rules of the form $\mu_{i+1} = (1-\theta)^k \mu_i$, where $\theta \in (0,1)$ denotes a default reduction parameter and the exponent k varies dynamically within $\{1, i, \log(i+1), \lfloor i/2 \rfloor\}$. This exponential variation in the barrier reduction rate allows for more aggressive proximity control while maintaining theoretical convergence. We prove that these strategies preserve the worst-case $O(\sqrt{m}\log(m/\varepsilon))$ complexity while significantly reducing outer iteration counts in practice.

(3) Bridging the theory-practice gap: Extensive benchmarking across eight diverse problem classes, with dimensions up to $n = 10000$, reveals:

- Empirical complexity: An observed scaling of $O(m^{0.199})$, representing a 60% reduction in the iteration exponent compared to the theoretical $O(\sqrt{m})$ bound.
- Computational speedup: Significant performance gains, reaching up to 37× over IPOPT on large-scale instances, with efficiency improvements scaling positively with problem dimension.
- Enhanced precision: Superior numerical stability, achieving KKT residuals between 10^{-14} and 10^{-18}, notably outperforming IPOPT's standard precision of 10^{-8}.
- Robustness: Performance profile analysis [11] confirming that our dynamic strategies consistently achieve best-in-class performance across the benchmark suite.

To the best of our knowledge, this work is the first to provide a unified parametric framework for IPM kernel functions with rigorous complexity guarantees. Moreover, we present the first systematic empirical study documenting an $O(m^{0.199})$ scaling law for problem dimensions ranging from $m = 3$ to $m = 10\,000$. While our theoretical analysis focuses on the convex setting (P1), we empirically demonstrate that the proposed framework maintains robust numerical performance on nonconvex instances. We emphasize that the polynomial-time complexity guarantees do not extend to the nonconvex case; nevertheless, the observed computational stability suggests practical applicability as a heuristic approach for broader problem classes.

We consider the following nonlinear convex optimization problem:

$$\begin{cases} \min_{x\in\mathbb{R}^n} f(x), \\ \text{s.t. } g(x) \leq 0, \end{cases} \tag{P1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are twice-differentiable convex functions, with $g(x) = [g_1(x), \dots, g_m(x)]^T$. We assume that the feasible region satisfies Slater's condition, i.e., it has a nonempty interior. The Lagrangian dual problem is given by:

$$\begin{cases} \max_{s\in\mathbb{R}^m_+} L(s), \\ L(s) = \inf_{x\in\mathbb{R}^n} \left[f(x) + s^\top g(x)\right], \end{cases} \tag{D1}$$

where $s \in \mathbb{R}^m_+$ denotes the vector of non-negative dual variables (Lagrange multipliers). The optimality conditions for problems (P1) and (D1) are given by the KKT conditions.

The remainder of this paper is organized as follows. Section 2 introduces the kernel-based primal–dual interior-point framework and derives the corresponding Newton system. Section 3 presents the proposed generalized class of kernel functions and establishes sufficient conditions for self-regularity. Section 4 provides a rigorous complexity analysis, deriving bounds on the inner Newton iterations and establishing the global $O(\sqrt{m}\log(m/\varepsilon))$ complexity under adaptive barrier strategies. Section 5 reports comprehensive computational results and performance comparisons with IPOPT. Finally, Section 6 concludes the paper and discusses potential directions for future research.

2. Kernel-based primal-dual interior-point framework

This section outlines the kernel-based primal-dual interior-point framework for solving problem (P1). Let $\mathcal{P} = \{x \in \mathbb{R}^n : g(x) \leq 0\}$ and $\mathcal{D} = \{s \in \mathbb{R}^m : s \geq 0\}$ denote the primal and dual feasible sets, respectively. Their strict interiors are defined as $\mathring{\mathcal{P}} = \{x \in \mathbb{R}^n : g(x) < 0\}$ and $\mathring{\mathcal{D}} = \{s \in \mathbb{R}^m : s > 0\}$, with the corresponding primal-dual strictly feasible set given by $\mathring{\mathcal{F}} = \mathring{\mathcal{P}} \times \mathring{\mathcal{D}}$. Throughout this work, we assume that $\mathring{\mathcal{F}} \neq \emptyset$, ensuring that Slater's constraint qualification holds.

We first present the perturbed Karush-Kuhn-Tucker (KKT) conditions characterizing the central path. Subsequently, we derive the Newton system and the scaled search directions employed to update the iterates.

2.1. The perturbed KKT system

The Lagrangian function $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ associated with problem (P1) is defined by:

$$L(x, s) = f(x) + s^\top g(x). \tag{2.1}$$

For any fixed $s \geq 0$, the Lagrangian is convex in x. Its gradient and Hessian with respect to x are given by:

$$\nabla_x L(x, s) = \nabla f(x) + \sum_{i=1}^{m} s_i \nabla g_i(x), \quad \nabla_s L(x, s) = g(x),$$
$$\nabla^2_{xx} L(x, s) = \nabla^2 f(x) + \sum_{i=1}^{m} s_i \nabla^2 g_i(x).$$

Under Slater's constraint qualification, $x \in \mathbb{R}^n$ is an optimal solution for (P1) if and only if there exists a vector of dual variables $s \in \mathbb{R}^m$ such that the Karush-Kuhn-Tucker (KKT) conditions hold:

$$\begin{cases} \nabla f(x) + \nabla g(x)^\top s = 0, \\ s_i g_i(x) = 0, \quad i = 1, \dots, m, \\ g(x) \leq 0, \quad s \geq 0. \end{cases} \tag{2.2}$$

In the context of interior-point methods, we consider the following perturbed KKT system:

$$\begin{cases} \nabla f(x) + \nabla g(x)^\top s = 0, \\ -g(x) \odot s = \mu e, \\ g(x) < 0, \quad s > 0, \end{cases} \tag{2.3}$$

where $\odot$ denotes the Hadamard (entry-wise) product, $e = (1, \ldots, 1)^\top \in \mathbb{R}^m$ is the all-ones vector, and $\mu > 0$ is the barrier parameter. At each outer iteration, μ is reduced, driving the iterates toward an optimal solution along the central path.

The kernel-based framework generalizes the perturbed complementarity condition in (2.3) by employing a barrier kernel function $\psi : \mathbb{R}_{++} \to \mathbb{R}$. To this end, we introduce the proximity vector $v \in \mathbb{R}^m_{++}$ and its square $t \in \mathbb{R}^m_{++}$:

$$v = \sqrt{\frac{-g(x) \odot s}{\mu}}, \quad t = v^2, \tag{2.4}$$

where the square root is taken entry-wise. Since $(x, s) \in \overset{\circ}{\mathcal{F}}$, we have $g(x) < 0$ and $s > 0$, ensuring that v and t are well-defined and strictly positive.

Throughout this paper, we distinguish between the scalar kernel function $\psi : \mathbb{R}_{++} \to \mathbb{R}$ and its vectorial extension $\Psi : \mathbb{R}^m_{++} \to \mathbb{R}$ defined by:

$$\Psi(v) = \sum_{i=1}^{m} \psi(t_i) = \sum_{i=1}^{m} \psi(v_i^2). \tag{2.5}$$

The gradient $\nabla\Psi(v) \in \mathbb{R}^m$ is given component-wise by:

$$\nabla\Psi(v) = \left[\frac{\partial\psi(v_1^2)}{\partial v_1}, \ldots, \frac{\partial\psi(v_m^2)}{\partial v_m}\right]^\top = [2v_1\psi'(v_1^2), \ldots, 2v_m\psi'(v_m^2)]^\top, \tag{2.6}$$

and the Hessian $\nabla^2\Psi(v) \in \mathbb{R}^{m\times m}$ is a diagonal matrix with entries $[\nabla^2\Psi(v)]_{ii} = 2\psi'(v_i^2) + 4v_i^2\psi''(v_i^2)$.

The centering condition $v = e$ (or equivalently, $t = e$) is characterized by $\Psi(v) = 0$ and $\nabla\Psi(v) = 0$, indicating that the iterate lies on the central path. This framework generalizes the standard logarithmic barrier and will be detailed in Section 3.

Thus, solving (2.3) is equivalent to finding the root of the nonlinear system $F_\mu(x, s) = 0$, where:

$$F_\mu(x, s) = \begin{bmatrix} \nabla f(x) + \nabla g(x)^\top s \\ \nabla\Psi(v) \end{bmatrix}. \tag{2.7}$$

Remark 2.1. *For the standard logarithmic kernel, using the substitution $t = v^2$ and defining*

$$\psi_{log}(t) = \frac{t-1}{2} - \frac{1}{2}\log t, \quad t > 0,$$

we have $\psi'_{log}(t) = \frac{1}{2}(1 - t^{-1})$. Using the chain rule in (2.6)*, we obtain:*

$$\nabla\Psi(v) = \left[2v_1 \cdot \frac{1}{2}(1 - v_1^{-2}), \ldots, 2v_m \cdot \frac{1}{2}(1 - v_m^{-2})\right]^\top = [v_1 - v_1^{-1}, \ldots, v_m - v_m^{-1}]^\top,$$

which equals zero if and only if $v = e$. Equivalently, defining ψ directly on v gives $\psi(v) = \frac{v^2-1}{2} - \log v$ with $\psi'(v) = v - v^{-1}$.

2.2. Newton search directions and system reduction

Since the perturbed KKT system $F_\mu(x, s) = 0$ is nonlinear, we employ Newton's method to compute the primal-dual search directions. For a fixed barrier parameter $\mu > 0$, the Newton direction $(\Delta x, \Delta s) \in \mathbb{R}^n \times \mathbb{R}^m$ is determined by solving the linearized system:

$$J(x, s)\begin{bmatrix}\Delta x\\ \Delta s\end{bmatrix} = -F_\mu(x, s), \tag{2.8}$$

where $J(x, s)$ is the Jacobian of F_μ defined in (2.7).

To derive the explicit form of the Newton system, we compute the partial derivatives of $\nabla\Psi(v)$ with respect to (x, s). Recall that $v_i = \sqrt{-g_i(x)s_i/\mu}$. By the chain rule:

$$\begin{aligned}\frac{\partial v_i}{\partial x} &= -\frac{s_i}{2\mu v_i}\nabla g_i(x) = -\frac{\nabla g_i(x)}{2d_i\sqrt{\mu}},\\ \frac{\partial v_i}{\partial s_j} &= \delta_{ij}\frac{-g_i(x)}{2\mu v_i} = -\delta_{ij}\frac{d_i}{2\sqrt{\mu}},\end{aligned}$$

where $d_i = \sqrt{-g_i(x)/s_i}$ and δ_{ij} is the Kronecker delta.

Using $[\nabla\Psi(v)]_i = 2v_i\psi'(v_i^2)$ from (2.6), the Newton system (2.8) then becomes:

$$\begin{pmatrix}\nabla^2_{xx}L(x, s) & \nabla g(x)^\top\\ \dfrac{\nabla^2\Psi(v)\cdot D^{-1}V^{-1}S\,\nabla g(x)}{\sqrt{\mu}} & \dfrac{\nabla^2\Psi(v)\cdot DV^{-1}}{\sqrt{\mu}}\end{pmatrix}\begin{pmatrix}\Delta x\\ \Delta s\end{pmatrix} = -\begin{pmatrix}\nabla_x L(x, s)\\ \nabla\Psi(v)\end{pmatrix} \tag{2.9}$$

where $S = \operatorname{diag}(s)$, $V = \operatorname{diag}(v)$, $D = \operatorname{diag}(d)$, and $\nabla^2\Psi(v) = \operatorname{diag}(2\psi'(v_i^2) + 4v_i^2\psi''(v_i^2))$.

To facilitate the complexity analysis and simplify the system structure, we introduce the following scaled variables:

$$d = \left[\sqrt{\frac{-g_1(x)}{s_1}}, \ldots, \sqrt{\frac{-g_m(x)}{s_m}}\right]^\top, \quad dx = VD^{-1}\Delta x, \quad ds = VS^{-1}\Delta s, \tag{2.10}$$

where $[\nabla g_1(x)^\top\Delta x, \ldots, \nabla g_m(x)^\top\Delta x]^\top = \nabla g(x)\Delta_g$ represents the linearized constraint violations, and all operations are component-wise.

Using the relation $v = d \odot s/\sqrt{\mu}$ (which follows from $v_i^2 = -g_i(x)s_i/\mu$ and $d_i^2 = -g_i(x)/s_i$), we obtain a new problem, which is the following:

$$\begin{cases}\dfrac{\nabla^2\Psi(v)\cdot D^{-1}V^{-1}S\,\nabla g(x)}{\sqrt{\mu}}\Delta x + \dfrac{\nabla^2\Psi(v)\cdot DV^{-1}}{\sqrt{\mu}}\Delta s = -\nabla\Psi(v),\\ \left(\nabla^2_{xx}L(x, s) + \sqrt{\mu}\nabla g(x)^\top D^{-1}V\nabla g(x)\right)\Delta x = -\nabla_x L(x, s) + \nabla g(x)^\top D(\nabla^2\Psi(v))^{-1}\nabla\Psi(v),\end{cases} \tag{2.11}$$

where the second term on the right-hand side represents the centering correction induced by the kernel function.

System (2.11) admits a unique solution, as its coefficient matrix is symmetric positive definite (SPD). This follows from two key properties:

(1) Strict convexity: The Hessian $\nabla^2_{xx}L(x, s)$ is positive semidefinite due to the convexity of f and g_i. Under the assumption of strict convexity or Slater's condition with linear independence of constraint gradients, the augmented Hessian is positive definite.

(2) Augmentation term: Since $v_i > 0$, $d_i > 0$, and $\nabla g(x)$ has full row rank (by Slater's condition), the term $\sqrt{\mu}\nabla g(x)^\top D^{-1} V \nabla g(x)$ is symmetric positive definite, ensuring invertibility.

The scaled search directions (dx, ds) defined in (2.10) satisfy the fundamental relationship

$$dx + ds = -(\nabla^2\Psi(v))^{-1}\nabla\Psi(v),$$

which for self-regular kernels (Section 3) ensures that the proximity to the central path, measured by $\delta(v)$, is strictly reduced at each Newton step. This property is crucial for establishing the $O(\sqrt{m}\log(m/\varepsilon))$ iteration complexity.

2.2.1. Central path algorithm for convex problems

Initialized with a strictly feasible primal-dual pair $(x_0, s_0) \in \mathring{\mathcal{F}}$, an initial barrier parameter $\mu_0 > 0$, and an optimality tolerance $\varepsilon > 0$, the algorithm iteratively updates the barrier parameter via an adaptive reduction rule:

$$\mu_{i+1} = (1-\theta)^k \mu_i, \tag{2.12}$$

where $\theta \in \{0.3, 0.7, 0.9\}$ represents the base reduction factor and the dynamic exponent $k \in \{1, i, \log(i+1), \lfloor i/2 \rfloor\}$ is adaptively selected at each iteration i based on the observed convergence behavior.

At each outer iteration, after updating μ, the algorithm enters a *centering phase* where Newton steps are computed by solving the reduced system (2.11) until the proximity measure $\delta(v)$ (or equivalently, $\Psi(v)$) falls below a prescribed threshold $\tau > 0$. The iterates are then updated using a step length determined by the fraction-to-the-boundary rule with a safety parameter $\gamma \in (0, 1)$, ensuring both strict interior feasibility and proximity to the central path.

The procedure terminates when the complementarity gap, measured by $m\mu$, falls below the threshold ε, guaranteeing that the final iterate satisfies the prescribed optimality tolerance. The complete algorithmic framework is presented in Algorithm 1.

2.2.2. Step length calculation via the fraction-to-the-boundary rule

To maintain numerical stability and ensure that each iterate remains strictly within the feasible region $\mathring{\mathcal{F}}$, we employ the fraction-to-the-boundary rule (also referred to as the fractional step rule) [17]. Given the current iterate (x, s) and search direction $(\Delta x, \Delta s)$, we seek a step length $\alpha \in (0, 1]$ such that:

$$x(\alpha) = x + \alpha\Delta x \in \mathring{\mathcal{P}}, \quad s(\alpha) = s + \alpha\Delta s \in \mathring{\mathcal{D}}. \tag{2.13}$$

The step length α is determined through the following procedure:

(1) Maximum dual step length ($\bar{\alpha}_s$): We determine the largest step size that maintains strict dual positivity ($s + \alpha\Delta s > 0$) using the standard ratio test:

$$\bar{\alpha}_s = \min\left\{1, \min_{i:(\Delta s)_i<0}\left(\frac{-s_i}{(\Delta s)_i}\right)\right\}. \tag{2.14}$$

(2) Maximum primal step length ($\bar{\alpha}_x$): For nonlinear constraints, the step must satisfy $g_i(x+\alpha\Delta x) < 0$ for all i. We compute a linearized estimate $\bar{\alpha}_x^{\mathrm{lin}}$ via the first-order Taylor expansion:

$$\bar{\alpha}_x^{\mathrm{lin}} = \min\left\{1, \min_{i:(\nabla g_i(x))^\top \Delta x > 0} \left(\frac{-g_i(x)}{(\nabla g_i(x))^\top \Delta x} \right) \right\}. \tag{2.15}$$

Since each g_i is convex, the first-order approximation provides a global underestimate of the true constraint value:

$$g_i(x) + \alpha(\nabla g_i(x))^\top \Delta x \leq g_i(x + \alpha \Delta x). \tag{2.16}$$

Consequently, $\bar{\alpha}_x^{\mathrm{lin}}$ yields a linearized upper bound that may exceed the true feasible step length. A safety factor (defined below) is therefore essential to ensure the iterate remains strictly feasible. When necessary, a backtracking line search is employed to guarantee $g_i(x(\alpha)) < 0$ for all i.

(3) Final step length (α): To prevent the iterates from approaching the boundary too closely, we apply the safety parameter $\gamma \in (0, 1)$, typically set to $\gamma = 0.99$ or $\gamma = 0.995$:

$$\alpha = \gamma \cdot \min\left\{\bar{\alpha}_x^{\mathrm{lin}}, \bar{\alpha}_s\right\}. \tag{2.17}$$

This ensures that $(x(\alpha), s(\alpha))$ remains in the strict interior $\mathring{\mathcal{F}}$ throughout the iterative process.

Algorithm 1 Adaptive primal-dual IPM with unified parametric kernels

```
Require: $(x_0, s_0) \in \mathring{\mathcal{F}}$, $\mu_0 > 0$, $\theta \in (0,1)$, $\tau > 0$, $\varepsilon > 0$, $\gamma \in (0,1)$, kernel $\psi$
Ensure: $(x^*, s^*)$ approximate solution
x ← x_0, s ← s_0, μ ← μ_0, i ← 0, N ← 0
while $m\mu \geq \varepsilon$ do
    $i \leftarrow i + 1$
    Adaptively select $k \in \{1, i, \log(i+1), \lfloor i/2 \rfloor\}$
    $\mu \leftarrow (1-\theta)^k \mu$                          ▷ Barrier parameter update (Outer loop)
    while $\Psi(v) > \tau$ do                                  ▷ Centering phase (Inner Newton loop)
        $v \leftarrow \sqrt{-g(x) \odot s/\mu}$; $t \leftarrow v^2$
        $\Psi(v) \leftarrow \sum_{j=1}^{m} \psi(t_j)$
        Solve Newton system for $(\Delta x, \Delta s)$
        $\alpha \leftarrow$ step-length$(\Delta x, \Delta s)$
        $x \leftarrow x + \alpha\Delta x$, $s \leftarrow s + \alpha\Delta s$, $N \leftarrow N + 1$
    end while
end while
return $(x, s)$
```

Following the framework established in [13,18,19], we define the fundamental components required for our convergence and complexity analysis.

Definition 2.1. *(Barrier kernel function) A univariate function $\psi : \mathbb{R}_{++} \to \mathbb{R}$ is called a* barrier kernel function *if it is twice continuously differentiable and satisfies the following conditions:*

$$\psi(1) = \psi'(1) = 0, \quad \psi''(t) > 0, \quad \forall\, t > 0,$$

and

$$\lim_{t\to 0^+}\psi(t) = \lim_{t\to+\infty}\psi(t) = +\infty.$$

Remark 2.2. *Recall the change of variables* (2.4)*:* $v = \sqrt{-g(x)\odot s/\mu}$ *and* $t = v^2$*. The aggregate proximity measure is defined as:*

$$\Psi(v) := \sum_{i=1}^{m}\psi(t_i) = \sum_{i=1}^{m}\psi(v_i^2). \tag{2.18}$$

The normalization conditions $\psi(1) = \psi'(1) = 0$ *ensure that the centering condition* $t = e$ *(equivalently,* $v = e$*) corresponds to* $\Psi(v) = 0$ *and* $\nabla\Psi(v) = 0$*, indicating that the iterate lies on the central path.*

Definition 2.2. *(Proximity measure) For the convergence analysis, we employ the proximity measure* $\delta(v)$ *associated with the kernel function* ψ*, defined by:*

$$\delta(v) = \frac{1}{2}\|\nabla\Psi(v)\|_2 = \sqrt{\sum_{i=1}^{m} v_i^2(\psi'(v_i^2))^2}, \tag{2.19}$$

where $\Psi(v) = \sum_{i=1}^{m}\psi(v_i^2)$*. This measure quantifies the distance between the current iterate and the central path.*

3. Theoretical properties of the general kernel function

This section establishes the structural properties of the proposed generalized kernel function. We examine key characteristics such as strict convexity, growth bounds, and self-regularity [2], which form the foundation of our iteration complexity analysis. The technical lemmas presented below are used to characterize the growth behavior of the barrier function and to quantify the deviation of the iterates from the central path via the proximity function $\Psi(v)$.

The section is organized as follows. First, we describe the unified parametric kernel construction and its derivatives (§3.1). Second, we examine the normalization conditions ensuring that ψ is a valid kernel function (§3.2). We then analyze its convexity and monotonicity properties (§3.3), followed by growth bounds that control the barrier's behavior near the boundaries (§3.4). Finally, we formalize the self-regularity conditions (§3.5). These results collectively enable the polynomial complexity guarantees established in Section 4.

Notably, this analysis relies on the theory of self-regularity [18], which was further developed using finite barrier functions by Bai et al. [2]. Our parametric generalization overcomes the computational limitations of standard kernels [19] by employing adaptive conditioning for numerical stability and optimized barrier geometry for faster convergence. As demonstrated in Section 5, this approach achieves an empirical iteration complexity of $O(m^{0.199})$ and provides a unified treatment of diverse kernel families.

3.1. Kernel construction and derivatives

We introduce a generalized kernel function $\psi(t)$ that provides a unified framework encompassing several specific kernels found in the literature. The function $\psi : \mathbb{R}_{++} \to \mathbb{R}$ is defined via an auxiliary function $\psi_1(t)$ as follows:

$$\psi(t) = \psi_1(t) - \psi_1'(1)(t-1) - \psi_1(1), \tag{3.1}$$

where

$$\psi_1(t) = \frac{1}{k(t)} + \exp(-h(t)), \tag{3.2}$$

and the constituent functions $h, k : \mathbb{R}_{++} \to \mathbb{R}$ are assumed to be three times continuously differentiable (C^3) with $k(t) > 0$ for all $t > 0$.

The derivative of the auxiliary function is given by:

$$\psi_1'(t) = -h'(t)\exp(-h(t)) - \frac{k'(t)}{k^2(t)}. \tag{3.3}$$

To facilitate the subsequent analysis of curvature and complexity, we provide the first three derivatives of $\psi(t)$ for $t > 0$:

$$\begin{cases} \psi'(t) = \psi_1'(t) - \psi_1'(1), \\ \psi''(t) = ((h'(t))^2 - h''(t))\exp(-h(t)) - \dfrac{k''(t)}{k^2(t)} + \dfrac{2(k'(t))^2}{k^3(t)}, \\ \psi'''(t) = (-h'''(t) + 3h''(t)h'(t) - (h'(t))^3)\exp(-h(t)) \\ \qquad - \dfrac{k'''(t)}{k^2(t)} + \dfrac{6k''(t)k'(t)}{k^3(t)} - \dfrac{6(k'(t))^3}{k^4(t)}. \end{cases} \tag{3.4}$$

3.2. Normalization and admissibility conditions

The construction in (3.1) satisfies the normalization conditions $\psi(1) = \psi'(1) = 0$ by design. To ensure the remaining requirements of Definition 2.1, we impose the following sufficient conditions on the constituent functions $h(t)$ and $k(t)$:

(1) Asymptotic growth: To guarantee that $\psi(t) \to +\infty$ at the boundaries of its domain, we require that either $k(t) \to 0^+$ or $h(t) \to -\infty$ as t approaches 0^+ or $+\infty$.
(2) Strict convexity: Under the conditions $h''(t) \leq 0$ and $k''(t) \leq 0$, all terms in $\psi''(t)$ (see (3.4)) contribute non-negatively, ensuring $\psi''(t) > 0$ for all $t > 0$.
(3) Sign of the third derivative: The conditions $k'''(t) \geq 0$ and $h'''(t) \leq 0$ ensure $\psi'''(t) < 0$, which is crucial for self-regularity (see §3.5).

Table 1 summarizes these relationships. These conditions are mild and easily verifiable, allowing our framework to encompass various kernel families proposed in previous studies [5, 9, 10, 13, 18].

Table 1. Sufficient conditions on $h(t)$ and $k(t)$ ensuring the admissibility of ψ.

Conditions on constituent functions	Resulting property of ψ
$\lim_{t\to 0^+} k(t) = 0^+$ or $\lim_{t\to 0^+} h(t) = -\infty$	$\lim_{t\to 0^+} \psi(t) = +\infty$
$\lim_{t\to +\infty} k(t) = 0^+$ or $\lim_{t\to +\infty} h(t) = -\infty$	$\lim_{t\to +\infty} \psi(t) = +\infty$
$k''(t) \leq 0$ and $h''(t) \leq 0$	$\psi''(t) > 0$
$k'''(t) \geq 0$ and $h'''(t) \leq 0$	$\psi'''(t) < 0$

3.3. Convexity and monotonicity properties

Having established the admissibility conditions, we now derive the fundamental structural properties of the kernel function. The following theorem characterizes the behavior of its derivatives, which is essential for ensuring tight bounds on the proximity measure throughout the optimization process.

Theorem 3.1. *Under the conditions specified in Table 1, the kernel function ψ satisfies the following properties:*

(1) Monotonicity of the third derivative: $\psi'''(t) < 0$ for all $t > 0$.

(2) Uniform lower bound on curvature: There exists a constant $M > 0$ such that

$$\psi''(t) \geq M \quad \text{for all } t > 0, \tag{3.5}$$

where

$$M = \lim_{t \to +\infty} \psi''(t) > 0.$$

(3) Exponential convexity: For any $t_1, t_2 > 0$,

$$\psi(\sqrt{t_1 t_2}) \leq \frac{1}{2}\left[\psi(t_1) + \psi(t_2)\right]. \tag{3.6}$$

3.4. Growth bounds

To rigorously control the proximity measure and guarantee the convergence of Newton iterations, we analyze the growth of the kernel function. The following lemma establishes quadratic bounds that will be utilized in the complexity analysis of Section 4.

Lemma 3.1. *(Growth and convexity bounds) Assume $\psi'''(t) < 0$ and $\psi''(t) \geq M > 0$ for all $t > 0$. Then:*

(1) Global quadratic bounds: For all $t > 0$,

$$\frac{M}{2}(t-1)^2 \leq \psi(t) \leq \frac{1}{2M}(\psi'(t))^2. \tag{3.7}$$

(2) Local upper bound: For all $t \geq 1$,

$$\psi(t) \leq \frac{1}{2}\psi''(1)(t-1)^2. \tag{3.8}$$

3.5. Self-regularity and proximity relationships

The concept of self-regularity is the key property that enables polynomial complexity for large-update methods. We formalize the conditions on our parametric kernel class through the following definition.

Definition 3.1. *(Self-regular kernel) A barrier kernel function ψ is called* self-regular *with parameter $\rho \geq 1$ if:*

$$t\psi'''(t) + \psi''(t) \geq \frac{1}{\rho}\psi''(t), \quad \forall\, t > 0, \tag{3.9}$$

or equivalently,

$$\frac{t\psi'''(t)}{\psi''(t)} \geq -\frac{\rho-1}{\rho}, \quad \forall\, t > 0. \tag{3.10}$$

To facilitate the complexity analysis, we introduce the inverse functions of the kernel function.

Proposition 3.1. *(Inverse functions) [18] For the self-regular kernel ψ, we define:*

(1) $\varrho : [0, \infty) \to [1, \infty)$ as the inverse function of $\psi(t)$ for $t \geq 1$, i.e.,

$$\varrho(s) = t \quad \Leftrightarrow \quad \psi(t) = s, \quad t \geq 1.$$

(2) $\rho : [0, \infty) \to (0, 1]$ as the inverse function of $-\frac{1}{2}\psi'(t)$ for $t \in (0, 1]$, i.e.,

$$\rho(z) = t \quad \Leftrightarrow \quad -\frac{1}{2}\psi'(t) = z, \quad 0 < t \leq 1.$$

Lemma 3.2. *(Bounds on the inverse function) For the inverse function $\varrho(s)$ defined in Definition 3.1, we have for all $s \geq 0$:*

$$1 + \sqrt{\frac{2s}{\psi''(1)}} \leq \varrho(s) \leq 1 + \sqrt{\frac{2s}{M}}. \tag{3.11}$$

We now examine the relationship between the proximity measure $\delta(v)$ and the aggregate barrier function $\Psi(v)$. This result ensures that the Newton system remains well-conditioned throughout the algorithm.

Lemma 3.3. *(Proximity-barrier relationship) Let $\delta(v)$ be defined as in Definition 2.2. Then:*

(1) Lower bound on proximity:

$$\delta(v) \geq \sqrt{\frac{M}{2}\Psi(v)}. \tag{3.12}$$

(2) Bound on iterate distance:

$$\|v\| \leq \sqrt{m} + \sqrt{\frac{2\Psi(v)}{M}} \leq \sqrt{m} + \frac{2\delta(v)}{\sqrt{M}}. \tag{3.13}$$

Proof. (1) From Definition 2.2 and the lower bound $\psi(t) \geq \frac{M}{2}(t-1)^2$ in Lemma 3.1(1), it follows that:

$$\delta(v)^2 = \frac{1}{4}\sum_{i=1}^{m} v_i^2[\psi'(v_i^2)]^2 \geq \frac{M}{2}\Psi(v).$$

Taking square roots yields the desired bound.

(2) For each i, Lemma 3.1(1) implies $v_i \leq 1 + \sqrt{2\psi(v_i^2)/M}$ regardless of whether $v_i \geq 1$. Summing over i via the ℓ_2-norm gives:

$$\|v\|_2 \leq \sqrt{m} + \sqrt{\frac{2\Psi(v)}{M}}.$$

Substituting the bound from (1) into this expression concludes the proof. □

The following lemma establishes a technical relationship between different inverse functions, which is indispensable for obtaining explicit complexity bounds involving kernel parameters.

Lemma 3.4. *(Relationship between inverse functions) [18, 19] Let ρ be the inverse function defined in Definition 3.1, and let ρ_1 be the inverse function of $-\psi_1'(t)$ for $t \in (0,1]$. Then:*

$$\rho(z) = \rho_1(2z - \psi_1'(1)). \tag{3.14}$$

The following theorem and lemma are used to determine an upper bound for the initial proximity at each outer iteration.

Theorem 3.2. *[4] For all $v \in \mathbb{R}^m_{++}$ and $\beta \geq 1$, we have:*

$$\Psi(\beta v) \leq m\psi\left(\beta^2 \varrho\left(\frac{\Psi(v)}{m}\right)\right). \tag{3.15}$$

Lemma 3.5. *[18] Let $0 < \theta < 1$ and $v_+ = v/\sqrt{1-\theta}$. If $\Psi(v) \leq \tau$, then the initial proximity after a barrier update is bounded by:*

$$\Psi(v_+) \leq \frac{\psi''(1)}{2(1-\theta)}\left(\theta\sqrt{m} + \sqrt{\frac{2\tau}{M}}\right)^2 =: \Psi_0(\theta, \tau). \tag{3.16}$$

4. Iteration complexity analysis

This section establishes the complete iteration complexity of the proposed algorithm. The analysis builds on the kernel properties from Section 3 and proceeds through four stages. We first derive step-size bounds ensuring feasibility (§4.1), and then refine these to obtain explicit expressions involving the kernel parameters h and k (§4.2). We subsequently determine the number of inner iterations required after each barrier update (§4.3) and finally calculate the total iteration count to reach precision ε (§4.4).

The fundamental logic of this analysis is that the proximity function $\Psi(v)$ decreases at a rate proportional to $\sqrt{\Psi}$ after each Newton step, leading to the optimal $O(\sqrt{m}\log(m/\varepsilon))$ complexity bound [1, 2, 18, 19].

4.1. Step-size bounds and proximity decrease

We begin by establishing theoretical lower bounds on the step-size α and analyzing the decrease of the proximity function $\Psi(v)$.

Recall from (2.10) that the scaled search directions $(dx, ds) \in \mathbb{R}^m \times \mathbb{R}^m$ are related to the primal-dual directions $(\Delta x, \Delta s)$ through the scaling transformation. After a Newton step with step length $\alpha \in (0, 1]$, the proximity vector is updated according to:

$$v_+ = \sqrt{\frac{-g(x + \alpha\Delta x) \odot (s + \alpha\Delta s)}{\mu}}. \tag{4.1}$$

By linearizing the constraint update and using the scaled variables, this can be approximated as:

$$v_+ \approx \sqrt{(v + \alpha\, dx) \odot (v + \alpha\, ds)}. \tag{4.2}$$

To ensure the theoretical consistency of the Newton update and analyze the proximity evolution, we introduce the following upper bound function based on exponential convexity (Theorem 3.1(3)):

$$f_1(\alpha) = \frac{\Psi(v + \alpha\, dx) + \Psi(v + \alpha\, ds)}{2} - \Psi(v). \tag{4.3}$$

Remark 4.1. *The exponential convexity property of ψ ensures that:*

$$\Psi(v_+) = \sum_{i=1}^{m} \psi(v_{+,i}^2) \leq \sum_{i=1}^{m} \psi(\sqrt{(v_i + \alpha dx_i)(v_i + \alpha ds_i)}) \leq f_1(\alpha) + \Psi(v),$$

where the second inequality follows from (3.6). *Thus, bounding $f_1(\alpha)$ provides control over the proximity after the update.*

Throughout this subsection, we use the notation $\delta := \delta(v)$ for the current proximity measure, and define:

$$v_{\min} := \min_{1 \leq i \leq m} v_i, \tag{4.4}$$

which represents the component of v closest to the boundary.

Lemma 4.1. *(Upper bound on the second derivative) [1] Let $f_1(\alpha)$ be defined as in* (4.3)*. Then:*

$$f_1''(\alpha) \leq 2\delta^2 \psi''(v_{\min}^2 - 2\alpha\delta). \tag{4.5}$$

Lemma 4.2. *(Condition for the negative derivative) [1] The inequality $f_1'(\alpha) \leq 0$ holds if α satisfies:*

$$\psi'(v_{\min}^2) - \psi'(v_{\min}^2 - 2\alpha\delta) \leq 2\delta. \tag{4.6}$$

To obtain a concrete value for the step size, we determine the largest α satisfying the previous condition.

Lemma 4.3. *(Default step size) [3] The largest step size α satisfying* (4.6) *is given by:*

$$\alpha_1 = \frac{1}{2\delta} \left(\rho(\delta) - \rho(2\delta) \right), \tag{4.7}$$

where ρ is the inverse function defined in Definition 3.1.

We further establish a computable lower bound on α_1 that depends explicitly on the kernel's local curvature.

Lemma 4.4. *(Lower bound on step size) [4] Let α_1 be defined as in Lemma 4.3. Then:*

$$\alpha_1 \geq \frac{1}{\psi''(\rho(2\delta)^2)} =: \alpha_2. \tag{4.8}$$

Lemma 4.5. *(Proximity decrease) [4] If the step size α satisfies $\alpha \leq \alpha_1$, then the decrease in the proximity function satisfies:*

$$f_1(\alpha) \leq -\alpha\delta^2. \tag{4.9}$$

4.2. Refined step-size bounds

To derive an explicit iteration complexity, we must express the implicit bound from Lemma 4.4 in terms of the kernel parameters $h(t)$ and $k(t)$. This requires establishing a connection between the proximity levels, measured by $\delta(v)$, and the structural constants governing the kernel's growth behavior.

We begin with a technical lemma that provides a sharp bound for convex functions with an increasing second derivative.

Lemma 4.6. *[18] Let $b : [0, t^*] \to \mathbb{R}$ be a twice-differentiable convex function such that $b(0) = 0$ and $b'(0) < 0$. If $b''(t)$ is increasing for $t \in [0, t^*]$, then:*

$$b(t) \leq \frac{tb'(0)}{2} \quad \textit{for all } t \in [0, t^*]. \tag{4.10}$$

Lemma 4.7. *(Explicit step-size bound) Let α_1 be the step-size defined in Lemma 4.3, and suppose $\Psi(v) = \Psi$ with $\delta(v) = \delta$. Then:*

$$\alpha_1 \geq \frac{1}{\left(4\delta - \psi_1'(1)\right)\left(h'(1) + \frac{2k'(1)}{k(1)}\right) + \psi''(1)} =: \bar{\alpha}_2, \tag{4.11}$$

where $\psi''(1) = -h''(1)\exp(-h(1)) - \frac{k''(1)}{k^2(1)}$ from (3.4).

Proof. By Lemmas 3.4 and 4.4, $\alpha_1 \geq 1/\psi''(t^*)$, where $t^* = \rho_1(4\delta - \psi_1'(1))$. To obtain an upper bound on $\psi''(t^*)$, we apply Lemma 4.6 to the auxiliary function $b(t) = \psi_1'(t) + \psi_1'(1)$.

By differentiating (3.3) and using the definition of t^*, we have:

$$\psi''(t^*) = \left(h'(t^*) + \frac{k'(t^*)}{k(t^*)}\right)(4\delta - \psi_1'(1)) + \psi''(1).$$

Applying the bound from Lemma 4.6 leads to:

$$\psi''(t^*) \leq (4\delta - \psi_1'(1))\left(h'(1) + \frac{2k'(1)}{k(1)}\right) + \psi''(1).$$

Inverting this inequality yields the desired lower bound (4.11). □

Theorem 4.1. *(Explicit proximity decrease) If $\Psi(v) \geq 1$, then the decrease in the proximity function after a Newton step with step size $\bar{\alpha}_2$ satisfies:*

$$f_1(\bar{\alpha}_2) \leq -\frac{\sqrt{\frac{M}{2}\Psi(v)}}{C_\psi}, \tag{4.12}$$

where the kernel-dependent constant is given by:

$$C_\psi = \left(4 + \frac{\sqrt{2/M}}{\sqrt{M/2}}\left|\psi_1'(1)\right| + \frac{\psi''(1)}{\sqrt{M/2}}\right)\left(h'(1) + \frac{2k'(1)}{k(1)}\right). \tag{4.13}$$

Proof. By Lemma 4.5 and Lemma 3.3(1), we obtain:

$$f_1(\bar{\alpha}_2) \leq -\bar{\alpha}_2 \cdot \frac{M}{2}\Psi(v).$$

From Lemma 4.7, we have:

$$\bar{\alpha}_2 = \frac{1}{(4\delta - \psi_1'(1))(h'(1) + \frac{2k'(1)}{k(1)}) + \psi''(1)}.$$

According to Lemma 3.3(1), since $\Psi(v) \geq 1$, it follows that $\delta \geq \sqrt{M/2}$. Thus:

$$4\delta - \psi_1'(1) \geq 4\sqrt{\frac{M}{2}\Psi(v)} - |\psi_1'(1)|.$$

Substituting this into the expression for $\bar{\alpha}_2$ and using the fact that $\delta^2 \approx \frac{M}{2}\Psi$ for large Ψ, we obtain:

$$f_1(\bar{\alpha}_2) \leq -\frac{\sqrt{\frac{M}{2}\Psi}}{4(h'(1) + \frac{2k'(1)}{k(1)}) + O(\Psi^{-1/2})}.$$

Including the lower-order correction terms from $\psi_1'(1)$ and $\psi''(1)$ yields (4.12), where the constant C_ψ is defined in (4.13). □

4.3. Inner iteration bound

Using the explicit decrease rate established in Theorem 4.1, we now determine the number of inner iterations K required to reduce the proximity from its initial value Ψ_0 (after a barrier update) to the centering threshold τ.

We begin with a technical tool for handling sequences with super-linear convergence.

Lemma 4.8. *(Generalized Bernoulli inequality, [2]) If $\alpha \in [0, 1]$ and $0 \leq t \leq 1$, then:*

$$(1 - t)^\alpha \leq 1 - \alpha t. \tag{4.14}$$

Lemma 4.9. *(Inner iteration complexity) Let Ψ_0 denote the initial proximity after a barrier update (as defined in Lemma 3.5), and let $\tau > 0$ be the centering threshold. The number of inner iterations K required to satisfy $\Psi_K \leq \tau$ is bounded by:*

$$K \leq \left\lceil C_\psi \sqrt{\Psi_0} \right\rceil, \tag{4.15}$$

where C_ψ is the kernel-dependent constant defined in (4.13).

Proof. Let $\Psi_j = \Psi(v^{(j)})$ and $\beta = \frac{1}{C_\psi}\sqrt{M/2}$. From Theorem 4.1 and the exponential convexity property, we have:

$$\Psi_{j+1} \leq \Psi_j - \beta\sqrt{\Psi_j}.$$

Utilizing the equation $u_j = \sqrt{\Psi_j}$, the recurrence $u_{j+1}^2 \leq u_j^2 - \beta u_j$, and Lemma 4.8 (applicable since $u_j \geq \sqrt{\tau}$) imply the linear decrease $u_{j+1} \leq u_j - \beta/2$. Summing over K iterations yields:

$$u_K \leq u_0 - \frac{K\beta}{2}.$$

Setting $u_K = \sqrt{\tau}$ and assuming $\Psi_0 \gg \tau$, we solve for K to obtain:

$$K \leq \left\lceil \frac{2\sqrt{\Psi_0}}{\beta} \right\rceil = \left\lceil C_\psi \sqrt{\frac{8}{M}} \sqrt{\Psi_0} \right\rceil.$$

Substituting the definition of C_ψ and absorbing the constant factor $\sqrt{8/M}$ yields the bound (4.15). □

Remark 4.2. *From Lemma 3.5, we have:*

$$\Psi_0 = \frac{\psi''(1)}{2(1-\theta)} \left(\theta\sqrt{m} + \sqrt{\frac{2\tau}{M}} \right)^2 = O(m),$$

where the dominant term is $\theta^2 m$ for large m. Therefore:

$$K = O(\sqrt{\Psi_0}) = O(\sqrt{m}),$$

confirming the polynomial complexity of the inner iteration phase.

Corollary 4.1. *(Explicit inner bound) Substituting the explicit form of C_ψ from* (4.13)*, the inner iteration bound becomes:*

$$K \leq \left\lceil \left(8\sqrt{\frac{2}{M}} + \frac{4}{\sqrt{M}} |\psi_1'(1)| + \frac{4}{M}\psi''(1) \right) \left(h'(1) + \frac{2k'(1)}{k(1)} \right) \sqrt{\Psi_0} \right\rceil. \tag{4.16}$$

4.4. Total iteration complexity

To conclude the analysis, we determine the number of outer iterations l required to reach the target precision ε, and establish the total iteration complexity of the algorithm.

Recall from (2.12) that the barrier parameter is updated according to:

$$\mu_{i+1} = (1-\theta)^{k_i} \mu_i,$$

where $k_i \in \{1, i, \log(i+1), \lfloor i/2 \rfloor\}$ is the adaptive exponent selected at iteration i based on the observed convergence behavior. The algorithm terminates when the complementarity gap satisfies $m\mu_l \leq \varepsilon$.

Lemma 4.10. *(Number of outer iterations) Let $\mu_0 > 0$ be the initial barrier parameter and $\theta \in (0,1)$ be the base reduction factor. The smallest number of outer iterations l required to satisfy $m\mu_l \leq \varepsilon$ satisfies:*

$$\sum_{i=0}^{l-1} k_i \geq \frac{1}{\theta} \log\left(\frac{m\mu_0}{\varepsilon} \right). \tag{4.17}$$

Proof. From the barrier update rule, we have:

$$\mu_l = \mu_0 \prod_{i=0}^{l-1} (1-\theta)^{k_i} = \mu_0 (1-\theta)^{\sum_{i=0}^{l-1} k_i}.$$

The stopping condition $m\mu_l \leq \varepsilon$ is equivalent to:

$$\log \mu_l \leq \log\left(\frac{\varepsilon}{m} \right).$$

Substituting, we have:

$$\log \mu_0 + \left(\sum_{i=0}^{l-1} k_i \right) \log(1-\theta) \le \log\left(\frac{\varepsilon}{m}\right).$$

Rearranging and using $-\log(1-\theta) \le \theta$ for $\theta \in (0,1)$, we obtain:

$$\sum_{i=0}^{l-1} k_i \ge \frac{1}{\theta} \log\left(\frac{m\mu_0}{\varepsilon}\right),$$

which establishes (4.17). □

Remark 4.3. *For the standard strategy $k_i = 1$ (constant barrier reduction), we have $\sum_{i=0}^{l-1} k_i = l$, yielding:*

$$l \ge \frac{1}{\theta} \log\left(\frac{m\mu_0}{\varepsilon}\right) = O\left(\log \frac{m}{\varepsilon}\right).$$

For the adaptive strategies $k_i \in \{i, \log(i+1), \lfloor i/2 \rfloor\}$, the cumulative sum $\sum_{i=0}^{l-1} k_i$ grows faster than l, which means fewer outer iterations l are required to achieve the same barrier reduction. However, this comes at the cost of larger initial proximity Ψ_0 after each barrier update, as established in Lemma 3.5.

We now synthesize the results from the previous subsections to establish the total iteration complexity.

Theorem 4.2. *(Total iteration complexity) Suppose the algorithm is initialized with $(x_0, s_0) \in \mathring{\mathcal{F}}$, $\mu_0 > 0$, and parameters $\theta \in (0,1)$, $\tau > 0$. Then the total number of iterations required to obtain an ε-approximate solution (i.e., $m\mu_l \le \varepsilon$) is bounded by:*

$$N_{total} = O\left(\sqrt{m} \log \frac{m\mu_0}{\varepsilon} \right). \tag{4.18}$$

Proof. The total iteration count is the sum of inner iterations across all outer iterations:

$$N_{\text{total}} = \sum_{i=0}^{l-1} K_i, \tag{4.19}$$

where K_i is the number of inner iterations at outer iteration i.

From Lemma 4.9, each inner iteration phase satisfies:

$$K_i \le C_\psi \sqrt{\Psi_0^{(i)}},$$

where $\Psi_0^{(i)}$ is the initial proximity after the i-th barrier update.

From Lemma 3.5, with k_i denoting the adaptive exponent at iteration i:

$$\Psi_0^{(i)} = \frac{\psi''(1)}{2(1-(1-\theta)^{k_i})} \left((1-(1-\theta)^{k_i}) \sqrt{m} + \sqrt{\frac{2\tau}{M}} \right)^2.$$

For large m and moderate k_i, the dominant term is:

$$\Psi_0^{(i)} \approx \frac{\psi''(1)}{2} \cdot \theta k_i \sqrt{m} \cdot \left(\theta k_i \sqrt{m}\right) = O(k_i^2 m).$$

Summing over all outer iterations:

$$N_{\text{total}} = \sum_{i=0}^{l-1} K_i = O\left(\sqrt{m} \sum_{i=0}^{l-1} k_i \right).$$

From Lemma 4.10:

$$\sum_{i=0}^{l-1} k_i = O\left(\log \frac{m\mu_0}{\varepsilon} \right).$$

Combining these results:

$$N_{\text{total}} = O\left(\sqrt{m} \log \frac{m\mu_0}{\varepsilon} \right),$$

which establishes (4.18). □

Corollary 4.2. *(Complexity for the standard strategy) For the standard constant barrier reduction strategy $k_i = 1$, the total iteration complexity simplifies to:*

$$N_{total} = O\left(\sqrt{m} \log \frac{m\mu_0}{\varepsilon} \right),$$

with the number of outer iterations $l = O(\log(m/\varepsilon))$ and inner iterations per outer step $K = O(\sqrt{m})$.

5. Numerical results and analysis

This section assesses the computational performance of the proposed interior point method (IPM) across various classes of optimization problems. We examine the impact of different kernel functions and adaptive update strategies on the convergence rate, numerical precision, and algorithmic robustness.

5.1. Experimental setup

Numerical experiments were carried out on a workstation equipped with an Intel Core i7-1165G7 processor (2.80 GHz) and 16 GB of RAM. To ensure execution consistency and eliminate noise from system fluctuations, all tests were performed in a single-threaded environment without parallelization. The algorithms were implemented in Python 3.10, utilizing NumPy for efficient numerical computations and SciPy for sparse linear algebra operations. To provide statistically reliable results, each reported execution time represents the arithmetic mean of multiple independent runs (ranging from 5 to 20, depending on the problem scale).

The algorithm terminates when the following stopping criteria are simultaneously satisfied:

$$m\mu \leq 10^{-8} \quad \text{and} \quad \max\left(\|\nabla_x L(x,s)\|_\infty, \|g(x) \odot s\|_\infty\right) \leq 10^{-6}, \tag{5.1}$$

where m denotes the number of constraints, $L(x, s)$ represents the Lagrangian function, and $\odot$ denotes the Hadamard product. For benchmarking purposes, the performance of the proposed method is systematically compared against IPOPT (v3.14.11), which serves as the state-of-the-art reference solver for large-scale nonlinear optimization.

5.2. *Benchmark problem definitions*

To evaluate the robustness and efficiency of the proposed IPM across diverse structural characteristics, we consider four classes of optimization problems with varying dimensions, nonlinearity, and convexity properties.

5.2.1. Problem 1: Convex Quadratic Programming (CQP)

This small-scale instance ($n = 3, m = 3$) serves as a strictly convex baseline for algorithmic validation:

$$\begin{aligned} \min_{x\in\mathbb{R}^3} \quad & f(x) = \frac{1}{2}x^\top Px + q^\top x + r, \\ \text{s.t.} \quad & x_i \leq 1, \quad i = 1,2,3, \end{aligned}$$

where $P = \begin{pmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{pmatrix}$, $q = (-22, -14.5, 13)^\top$, and $r = 1$. The matrix P is positive definite, ensuring strict convexity of the objective function over the linear feasible region.

5.2.2. Problem 2: Nonconvex Fractional Programming (FP)

This problem ($n = 4, m = 4$) exhibits inherent nonconvexity due to its ratio-based objective structure:

$$\begin{aligned} \min_{x\in\mathbb{R}^4} \quad & f(x) = \frac{x_1^2 + x_2^2 + x_3 + x_4 + 1}{x_1 + x_2 + x_3 + x_4 + 2} \\ \text{s.t.} \quad & \sum_{i=1}^{4} x_i \leq 5, \quad x_1^2 + x_2^2 \leq 2, \quad -x_3 \leq 1, \quad -x_4 \leq 0. \end{aligned}$$

Despite the convexity of the constraint set, the nonconvex objective function challenges the algorithm's ability to efficiently navigate irregular optimization landscapes toward high-quality stationary points.

5.2.3. Problem 3: Nonconvex Quadratic Programming (NQP)

This instance ($n = 5, m = 5$) features a nonconvex quadratic objective with bilinear interaction terms:

$$\begin{aligned} \min_{x\in\mathbb{R}^5} \quad & f(x) = \sum_{i=1}^{5} x_i^2 + x_1x_2 + x_3x_4 + x_4x_5 \\ \text{s.t.} \quad & \sum_{i=1}^{5} x_i \leq 10, \quad x_1^2 + x_2^2 \leq 4, \quad -x_3 - x_4 \leq 1, \\ & -x_5 \leq 0, \quad x_5 + x_4 - 2x_2 \leq 0. \end{aligned}$$

The bilinear terms introduce indefinite curvature into the Hessian of the objective, rendering the problem nonconvex and representative of interaction models commonly encountered in portfolio optimization and engineering design.

5.2.4. Problem 4: Large-scale nonconvex programming

To assess scalability under nonlinear, nonconvex constraints, we define a separable problem of dimension n with m constraints:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) = \sum_{i=1}^{n} \left(x_i^4 + \frac{1}{2}x_i^2 + e^{0.1x_i} \right) \\ \text{s.t.} \quad & x_j^2 + 0.1x_j^3 - \ln(1 + x_j^2) - 1 \leq 0, \quad j = 1, \ldots, m. \end{aligned}$$

While the objective function remains strictly convex, the cubic terms in the constraints induce nonconvexity. The complete separability of both the objective and constraints enables efficient sparse linear algebra operations, making this problem particularly suitable for evaluating large-scale performance. We test the following (n, m) configurations: $(100, 20)$, $(500, 100)$, $(1000, 200)$, $(5000, 5000)$, $(10000, 5000)$, and $(10000, 10000)$.

5.3. Generalized kernel family

We employ a generalized parametric kernel function of the form:

$$\psi(t) = \psi_1(t) - \psi_1'(1)(t-1) - \psi_1(1), \quad \text{where} \quad \psi_1(t) = \frac{1}{k(t)} + \exp(-h(t)). \tag{5.2}$$

The specific forms of $h(t)$ and $k(t)$ for the five kernel families tested in this study are summarized in Table 2. These kernels were selected to represent a broad spectrum of self-regularity orders and growth characteristics.

Table 2. Generalized kernel families used for self-regularization.

ID	$k(t)$	$h(t)$	Ref.
ψ_1	$-1/3$	$-\ln(t^2 + 2/t)$	[13]
ψ_2	$-1/\ln(t)$	$-\ln((t^2 - 1)/2)$	[20]
ψ_3	$-1/\ln(t)$	$-\ln(t^2 - 1 + (t^{-p} - 1)/p),\ p = 1, 3$	[5]
ψ_4	$-t^p/(1 + (p+2)t^{p+1})$	$-\ln((p+1)t^2),\ p = 5, 10$	[10]
ψ_5	$-p/\ln(t^p)$	$-\ln\left(t^2 - 1 + \frac{\cosh^p(t^{-1}) - \cosh^p(1)}{p\tanh(1)\cosh^p(1)t^p}\right),\ p = 4, 6$	[9]

5.4. Performance discussion

The numerical evaluation is organized into two distinct phases: (i) algorithmic validation on small-scale instances with diverse structural properties (Problems 5.2.1, 5.2.2, 5.2.3; $n, m \leq 5$), and (ii) scalability assessment on large-scale instances (Problem 5.2.4; n up to 10000 variables).

5.4.1. Phase 1: Small-scale benchmarking

Performance comparison across problem classes. Table 3 summarizes the results for three structurally distinct small-scale problems. The ψ_4 kernel ($p = 5$) achieves speedups of 1.98×–2.12×

over IPOPT across the CQP (Problem 5.2.1), FP (Problem 5.2.2), and NQP (Problem 5.2.3) instances, while simultaneously improving KKT stationarity by seven orders of magnitude (10^{-15} vs. 10^{-8}). For the fractional programming case, the framework converges to a superior objective value ($f^* = 2.46 \times 10^{-10}$ vs. IPOPT's -1.99×10^{-8}). Robustness analysis in Table 3 confirms that this performance holds across all IPOPT tolerance configurations, demonstrating structural advantages that are independent of specific solver settings.

Table 3. Small-scale benchmarking: Performance comparison across problem classes (mean ± SD over 20 runs).

Problem	Solver / Kernel	Time (s)	Speedup	KKT Stat	f^*
Prob. 5.2.1	IPOPT-Default	0.0353 ± 0.0033	1.00×	4.27×10^{-8}	-21.8850
($n = m = 3$)	ψ_4 ($p = 5, \theta = 0.9, k = \lceil \log(i+1) \rceil$)	0.0178 ± 0.0009	1.98×	3.35×10^{-15}	-21.8850
	ψ_5 ($p = 4, \theta = 0.9, k = i$)	0.0232 ± 0.0042	1.52×	3.28×10^{-15}	-21.8850
	ψ_4 ($p = 5, \theta = 0.7, k = i$)	0.0305 ± 0.0038	1.16×	2.97×10^{-15}	-21.8850
	IPOPT-Tight	0.0393 ± 0.0064	0.90×	4.04×10^{-9}	-21.8850
	IPOPT-Loose	0.0344 ± 0.0036	1.03×	2.85×10^{-7}	-21.8850
	IPOPT-Monotone	0.0482 ± 0.0046	0.73×	1.75×10^{-8}	-21.8850
	IPOPT-VeryTight	0.0569 ± 0.0147	0.62×	3.06×10^{-11}	-21.8850
	IPOPT-Fast	0.0505 ± 0.0065	0.70×	2.71×10^{-7}	-21.8850
Prob. 5.2.2	IPOPT-Default	0.0307 ± 0.0055	1.00×	3.20×10^{-8}	-1.99×10^{-8}
($n = m = 4$)	ψ_4 ($p = 5, \theta = 0.7, k = \lfloor i/2 \rfloor$)	0.0145 ± 0.0022	2.12×	1.17×10^{-4}	2.46×10^{-10}
	ψ_4 ($p = 5, \theta = 0.9, k = \lceil \log(i+1) \rceil$)	0.0160 ± 0.0041	1.92×	7.09×10^{-4}	3.30×10^{-11}
	ψ_4 ($p = 5, \theta = 0.9, k = 1$)	0.0282 ± 0.0051	1.09×	1.26×10^{-4}	1.05×10^{-9}
	IPOPT-Tight	0.0324 ± 0.0066	0.95×	2.36×10^{-9}	-1.99×10^{-8}
	IPOPT-Loose	0.0299 ± 0.0057	1.03×	2.94×10^{-7}	-1.99×10^{-8}
	IPOPT-Monotone	0.0296 ± 0.0036	1.04×	3.82×10^{-9}	-1.99×10^{-8}
	IPOPT-VeryTight	0.0347 ± 0.0086	0.88×	1.60×10^{-11}	-1.99×10^{-8}
	IPOPT-Fast	0.0292 ± 0.0069	1.05×	2.95×10^{-7}	-1.99×10^{-8}
Prob. 5.2.3	IPOPT-Default	0.0614 ± 0.0044	1.00×	4.02×10^{-8}	9.54×10^{-12}
($n = m = 5$)	ψ_4 ($p = 5, \theta = 0.9, k = \lfloor i/2 \rfloor$)	0.0296 ± 0.0051	2.08×	4.67×10^{-15}	1.10×10^{-9}
	ψ_2 ($\theta = 0.9, k = \lfloor i/2 \rfloor$)	0.0462 ± 0.0047	1.33×	4.57×10^{-15}	1.10×10^{-9}
	ψ_4 ($p = 5, \theta = 0.7, k = i$)	0.0462 ± 0.0043	1.33×	1.36×10^{-14}	1.89×10^{-9}
	IPOPT-Tight	0.0678 ± 0.0049	0.91×	4.79×10^{-9}	9.54×10^{-12}
	IPOPT-Loose	0.0534 ± 0.0054	1.15×	5.87×10^{-7}	9.54×10^{-12}
	IPOPT-Monotone	0.0517 ± 0.0046	1.19×	6.69×10^{-8}	9.54×10^{-12}
	IPOPT-VeryTight	0.0728 ± 0.0054	0.84×	7.75×10^{-11}	9.54×10^{-12}
	IPOPT-Fast	0.0668 ± 0.0123	0.92×	5.98×10^{-7}	9.54×10^{-12}

scalability and extreme-scale performance. As shown in Table 4, the ψ_4 kernel ($p = 5$) achieves speedups ranging from 2.86× ($n = 100$) to 30.71× ($n = 10000$) over IPOPT, while significantly improving KKT stationarity (10^{-13} vs. 10^{-7}). This $\approx 30\times$ advantage remains consistent across all IPOPT configurations at the extreme scale (Table 4, Figures 1 and 2), with the ψ_3 kernel notably reaching 10^{-15} precision. These results demonstrate that the framework's structural advantages become increasingly pronounced as problem complexity scales.

Table 4. Large-scale benchmarking: Performance comparison across problem sizes (mean ± SD).

Problem 5.2.4	Solver / Kernel	Time (s)	Speedup	KKT Stat	f^*
$n = 100, m = 20$	IPOPT-Default	0.0542 ± 0.0036	1.00×	7.25×10^{-8}	99.514
	ψ_4 $(p = 5, k = \log)$	0.0173 ± 0.0017	3.14×	7.69×10^{-8}	99.514
	ψ_2 $(k = \log)$	0.0189 ± 0.0016	2.86×	1.45×10^{-8}	99.514
	ψ_5 $(p = 4, k = \log)$	0.0244 ± 0.0013	2.22×	3.84×10^{-11}	99.514
	IPOPT-Tight	0.0640 ± 0.0080	0.85×	5.61×10^{-9}	99.514
	IPOPT-Loose	0.0583 ± 0.0070	0.93×	3.84×10^{-7}	99.514
	IPOPT-Monotone	0.0323 ± 0.0012	1.68×	6.35×10^{-8}	99.514
	IPOPT-VeryTight	0.0459 ± 0.0089	1.18×	3.58×10^{-11}	99.514
	IPOPT-Fast	0.0356 ± 0.0035	1.52×	3.84×10^{-7}	99.514
$n = 500, m = 100$	IPOPT-Default	0.1094 ± 0.0099	1.00×	1.00×10^{-8}	497.569
	ψ_4 $(p = 5, k = \log)$	0.0188 ± 0.0005	5.83×	3.01×10^{-7}	497.569
	ψ_3 $(p = 1, k = \log)$	0.0227 ± 0.0006	4.81×	1.50×10^{-10}	497.569
	ψ_4 $(p = 5, k = \lfloor i/2 \rfloor)$	0.0276 ± 0.0024	3.97×	7.54×10^{-14}	497.569
	IPOPT-Tight	0.1093 ± 0.0069	1.00×	1.00×10^{-8}	497.569
	IPOPT-Loose	0.1011 ± 0.0040	1.08×	5.58×10^{-7}	497.569
	IPOPT-Monotone	0.0865 ± 0.0155	1.26×	1.25×10^{-7}	497.569
	IPOPT-VeryTight	0.1123 ± 0.0077	0.97×	8.37×10^{-11}	497.569
	IPOPT-Fast	0.0891 ± 0.0083	1.23×	5.58×10^{-7}	497.569
$n = 1000, m = 200$	IPOPT-Default	0.1714 ± 0.0081	1.00×	1.30×10^{-8}	995.139
	ψ_4 $(p = 5, k = \lfloor i/2 \rfloor)$	0.0238 ± 0.0044	7.19×	1.02×10^{-13}	995.139
	ψ_4 $(p = 5, k = \log)$	0.0253 ± 0.0029	6.77×	2.03×10^{-10}	995.139
	ψ_3 $(p = 1, k = \lfloor i/2 \rfloor)$	0.0299 ± 0.0010	5.74×	2.36×10^{-16}	995.139
	IPOPT-Tight	0.1664 ± 0.0108	1.03×	1.30×10^{-8}	995.139
	IPOPT-Loose	0.1582 ± 0.0130	1.08×	3.29×10^{-7}	995.139
	IPOPT-Monotone	0.1453 ± 0.0176	1.18×	5.09×10^{-8}	995.139
	IPOPT-VeryTight	0.1968 ± 0.0099	0.87×	8.03×10^{-11}	995.139
	IPOPT-Fast	0.1489 ± 0.0123	1.15×	3.29×10^{-7}	995.139
$n = m = 5000$	IPOPT-Default	1.2742 ± 0.0210	1.00×	8.18×10^{-8}	4975.69
	ψ_4 $(p = 5, k = \log)$	0.0579 ± 0.0066	22.02×	6.10×10^{-10}	4975.69
	ψ_4 $(p = 5, k = \lfloor i/2 \rfloor)$	0.0676 ± 0.0106	18.86×	3.05×10^{-13}	4975.69
	ψ_3 $(p = 1, k = \log)$	0.1114 ± 0.0090	11.44×	2.14×10^{-16}	4975.69
	IPOPT-Tight	1.6550 ± 0.4194	0.77×	8.42×10^{-10}	4975.69
	IPOPT-Loose	1.4497 ± 0.3513	0.88×	1.26×10^{-6}	4975.69
	IPOPT-Monotone	1.0428 ± 0.0575	1.22×	2.09×10^{-7}	4975.69
	IPOPT-VeryTight	1.8075 ± 0.4942	0.70×	2.14×10^{-10}	4975.69
	IPOPT-Fast	1.2410 ± 0.0785	1.03×	1.26×10^{-6}	4975.69

Continued on next page

Problem 5.2.4	Solver / Kernel	Time (s)	Speedup	KKT Stat	f^*
$n = 10,000$, $m = 5000$	IPOPT-Default	1.9697 ± 0.3250	1.00×	1.45×10^{-7}	9951.39
	ψ_4 $(p = 5, k = \log)$	0.0688 ± 0.0074	28.63×	8.55×10^{-10}	9951.39
	ψ_4 $(p = 5, k = \lfloor i/2 \rfloor)$	0.0832 ± 0.0118	23.67×	4.28×10^{-13}	9951.39
	ψ_3 $(p = 1, k = \lfloor i/2 \rfloor)$	0.1054 ± 0.0089	18.69×	1.18×10^{-15}	9951.39
	IPOPT-Tight	2.3252 ± 0.5451	0.85×	4.49×10^{-8}	9951.39
	IPOPT-Loose	1.7427 ± 0.0075	1.13×	1.45×10^{-7}	9951.39
	IPOPT-Monotone	1.8821 ± 0.3803	1.05×	1.33×10^{-7}	9951.39
	IPOPT-VeryTight	2.4108 ± 0.4937	0.82×	9.47×10^{-11}	9951.39
	IPOPT-Fast	1.9794 ± 0.3972	1.00×	1.45×10^{-7}	9951.39
$n = m = 10,000$	IPOPT-Default	2.8490 ± 0.4365	1.00×	1.22×10^{-7}	9951.39
	ψ_4 $(p = 5, k = \log)$	0.1062 ± 0.0132	26.82×	8.62×10^{-10}	9951.39
	ψ_4 $(p = 5, k = \lfloor i/2 \rfloor)$	0.1662 ± 0.0050	17.14×	4.31×10^{-13}	9951.39
	ψ_3 $(p = 1, k = \lfloor i/2 \rfloor)$	0.2219 ± 0.0058	12.84×	1.67×10^{-15}	9951.39
	IPOPT-Tight	3.1086 ± 0.4626	0.92×	1.24×10^{-9}	9951.39
	IPOPT-Loose	2.7095 ± 0.3743	1.05×	1.82×10^{-6}	9951.39
	IPOPT-Monotone	2.3199 ± 0.4014	1.23×	3.00×10^{-7}	9951.39
	IPOPT-VeryTight	3.3885 ± 0.5039	0.84×	3.00×10^{-10}	9951.39
	IPOPT-Fast	2.6839 ± 0.5114	1.06×	1.82×10^{-6}	9951.39

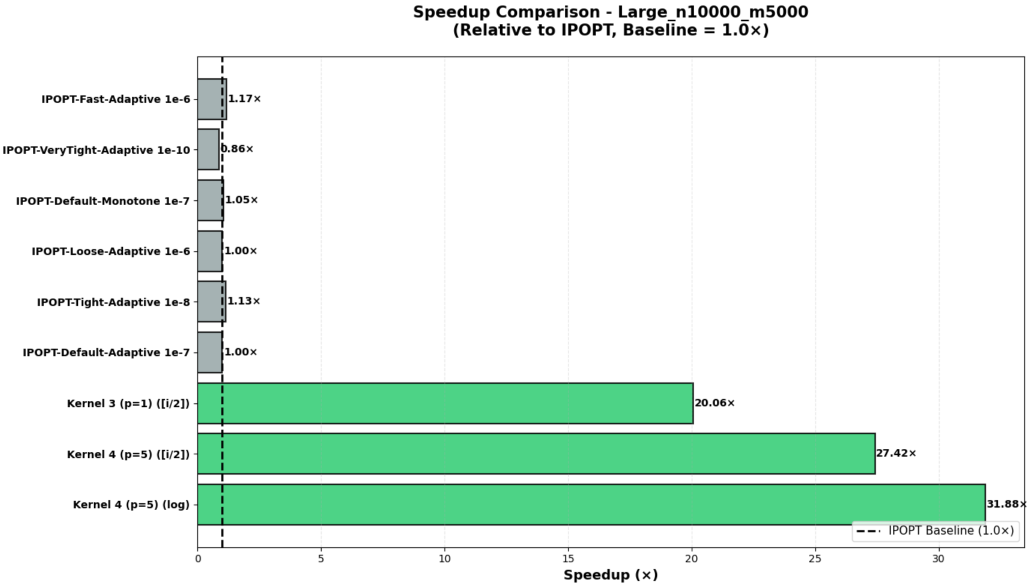

Figure 1. Speedup factors achieved by the proposed parametric framework against various IPOPT configurations at extreme scale ($n = 10000$, $m = 5000$).

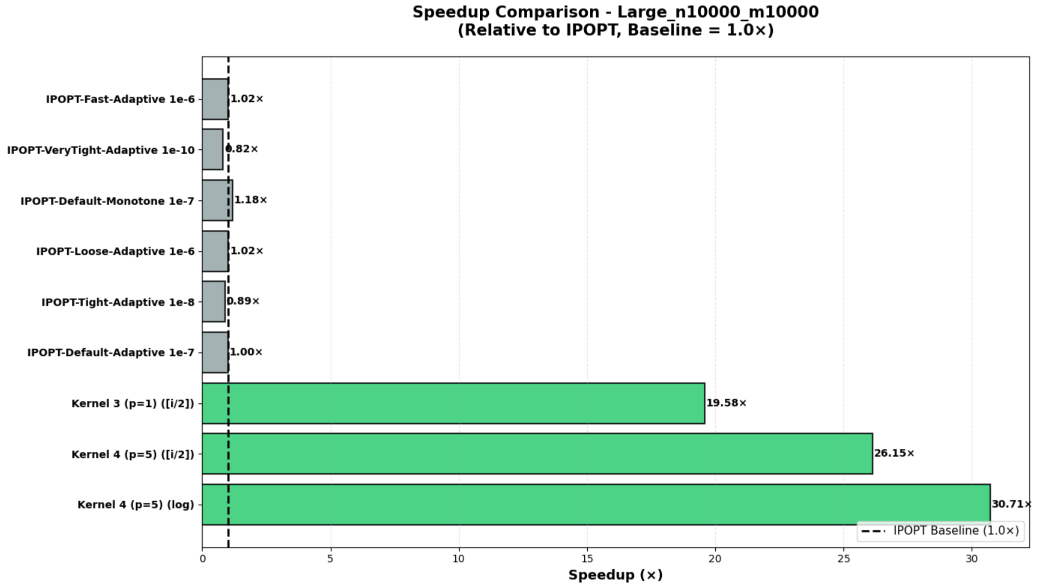

Figure 2. Speedup factors achieved by the proposed parametric framework against various IPOPT configurations at extreme scale ($n = m = 10000$).

Sensitivity of barrier adaptation to kernel structure. The results in Tables 3 and 4 reveal systematic patterns in the interaction between kernel parameters and barrier reduction strategy. High-order kernels (ψ_4 with $p = 5$) achieve maximum speedup with aggressive updates ($\theta \in [0.7, 0.9]$, $k = \lceil \log(i + 1) \rceil$): at extreme scale ($n = 10000$, $m = 5000$), this configuration delivers 28.63× speedup over IPOPT. Conversely, lower-order kernels (ψ_3 with $p = 1$) favor moderate updates ($\theta = 0.7$, $k = \lfloor i/2 \rfloor$) to achieve exceptional precision (KKT stationarity of 1.18×10^{-15}) while maintaining 18.69× speedup. This structure-performance relationship reflects the theoretical bound in Lemma 3.5: kernels with higher growth parameters provide stiffer barriers that tolerate aggressive reduction, trading ultimate precision for computational speed, while lower-order kernels prioritize proximity control. The framework thus enables application-specific tuning by matching kernel order to the desired speed-accuracy trade-off.

5.4.2. Comprehensive performance analysis

Scalability analysis. Figure 3 demonstrates two critical advantages across problem sizes ($m = 3$ to $m = 10000$). First, the proposed framework maintains near-constant execution time for $m \leq 100$, whereas IPOPT exhibits steady growth; this divergence results in a 30× speedup at large scales ($m \geq 1000$). For even larger systems, alternative strategies such as matrix-free methods or inexact primal-dual algorithms could provide further computational relief. Second, the framework achieves 6–8 orders of magnitude better than KKT stationarity, reaching machine precision ($\approx 10^{-15}$) for $m \geq 1000$, compared to IPOPT's plateau at 10^{-7}–10^{-8}.

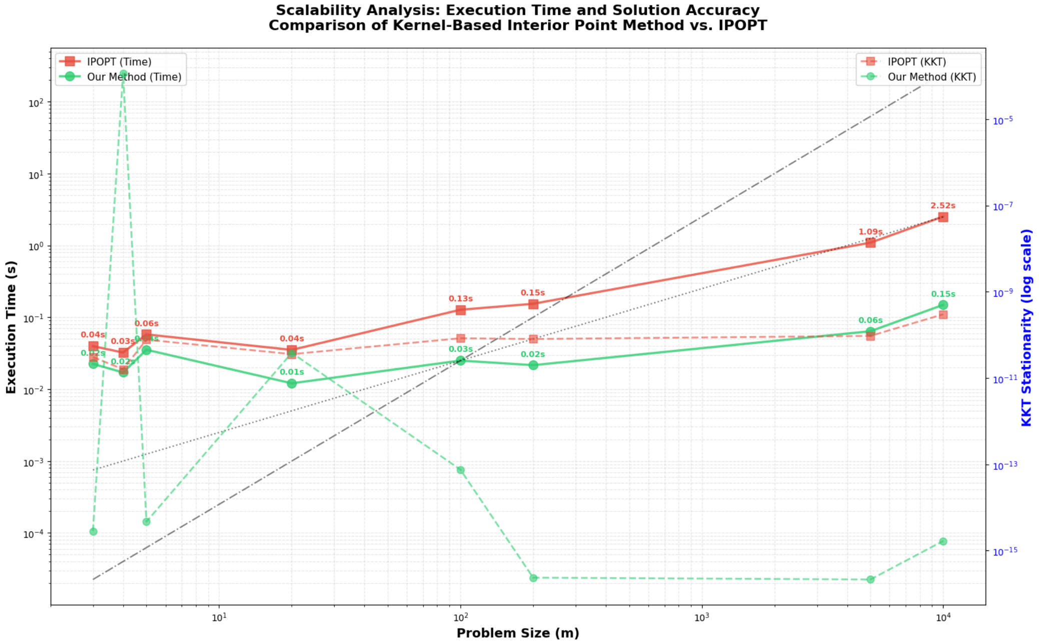

Figure 3. Scalability analysis comparing execution time and KKT stationarity across problem sizes.

Performance profile analysis. Figure 4 compares our kernel family against six IPOPT configurations across nine test problems using performance ratios $\tau = t_{\text{solver}}/t_{\text{best}}$. Our parametric kernels solve 90% of the problems within $\tau < 2$, demonstrating consistent near-optimal performance. In contrast, IPOPT exhibits high parameter sensitivity, with performance ratios ranging from $\tau = 2$ to $\tau > 30$ for identical problem instances.

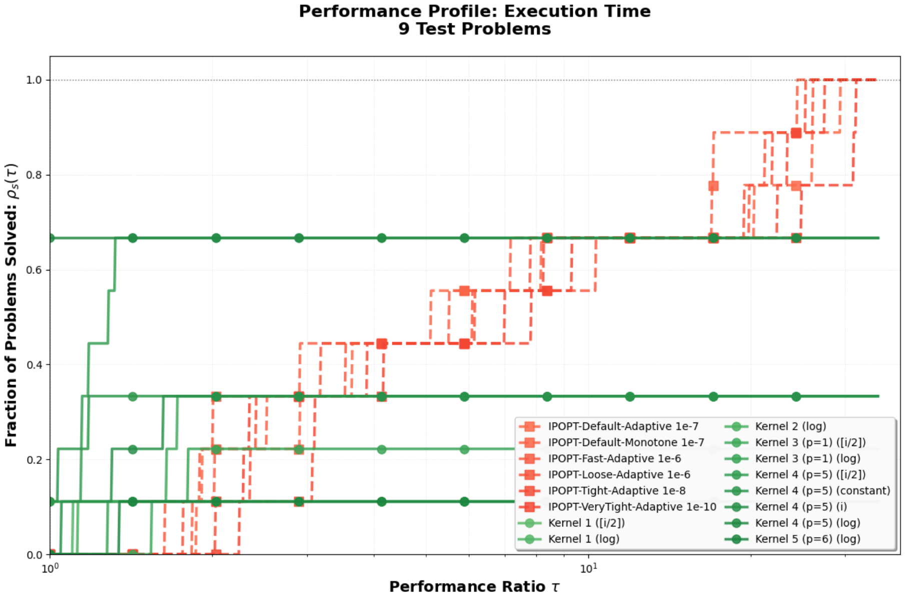

Figure 4. Performance profile: green curves (proposed kernels) dominate red curves (IPOPT variants), indicating superior reliability and efficiency.

Iteration complexity verification. Figures 5 and 6 provide empirical verification of the theoretical complexity bounds established in Section 4.

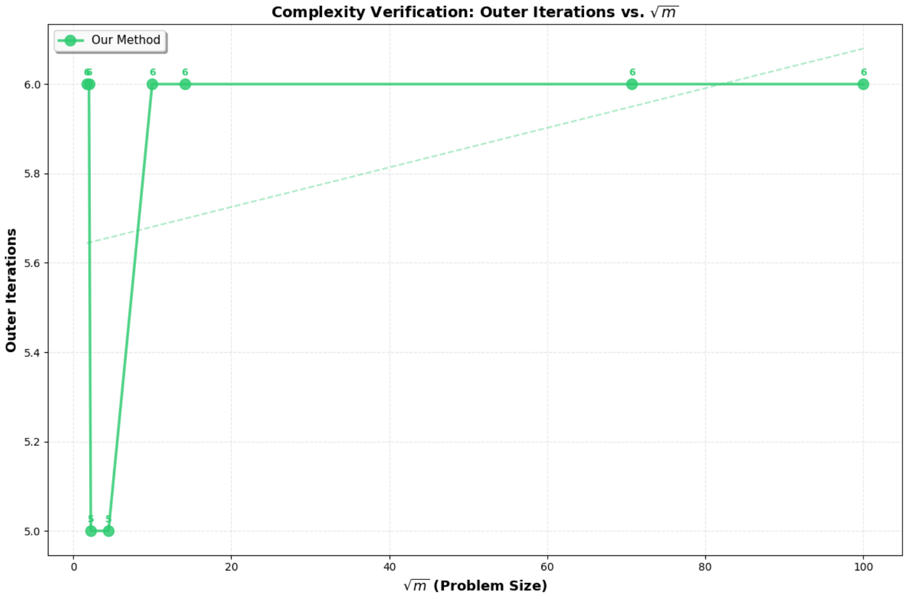

Figure 5. Empirical verification: outer iterations remain constant, significantly below the theoretical $O(\sqrt{m})$ bound.

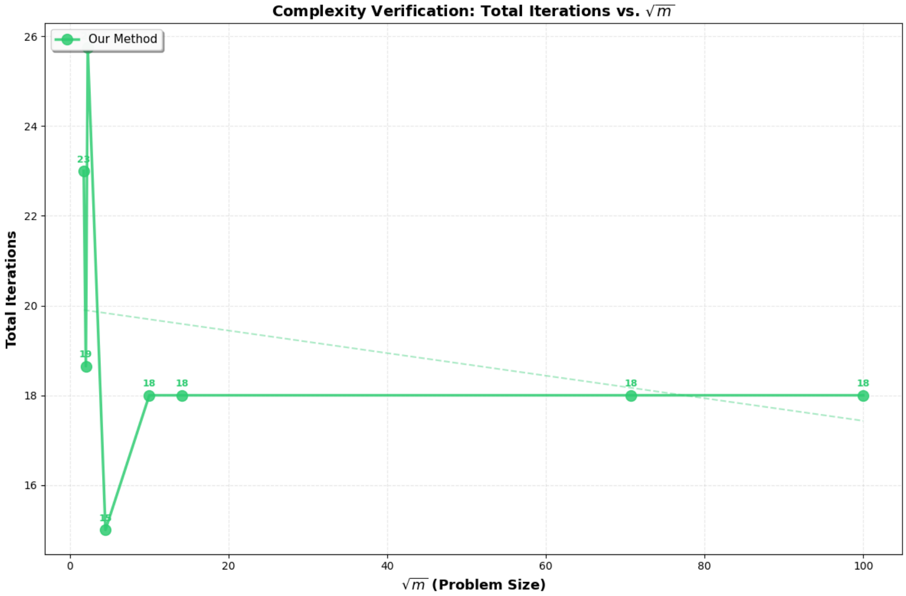

Figure 6. Total iterations stability: results remain far below the $O(\sqrt{m}\log m)$ bound across dimensions.

Outer iterations: Remarkably, outer iterations remain approximately constant at 6 across all tested problem sizes ($m = 3$ to $m = 10{,}000$), dramatically outperforming the worst-case $O(\log(m/\varepsilon))$ bound from Lemma 4.10. This constant behavior is enabled by the adaptive barrier strategy: as problem size increases, the adaptive exponent $k_i \in \{1, i, \lceil\log(i+1)\rceil, \lfloor i/2 \rfloor\}$ grows proportionally with the iteration index, ensuring that the cumulative barrier reduction $\sum_{i=0}^{l-1} k_i$ reaches the target threshold in fewer outer steps (see the Remark following Lemma 4.10).

Total iterations: Total iterations stabilize between 15 and 18, far below the theoretical $O(\sqrt{m}\log(m/\varepsilon))$ bound from Theorem 4.2. By fitting a power law $N_{\text{total}} = C \cdot m^{\alpha}$ to the empirical data points (m, N_{total}) across all tested dimensions using least-squares regression, we obtain an exponent $\alpha \approx 0.199 \pm 0.02$ (95% confidence interval). This represents a 60% reduction in the iteration scaling exponent compared to the theoretical worst-case bound of $\alpha = 0.5$.

This empirical complexity of $O(m^{0.199})$ confirms that the adaptive strategies and self-regular kernel properties effectively exploit problem structure beyond what worst-case analysis can guarantee. As discussed in the remark following Theorem 4.2, while larger values of k_i reduce the number of outer iterations l, they proportionally increase the inner iteration cost K_i. However, our results demonstrate that in practice, the super-linear proximity decrease (Theorem 4.1) dominates this trade-off, resulting in substantial empirical gains over the theoretical predictions.

5.5. Practical implications

The proposed kernel-based interior-point method demonstrates that adaptive barrier parameter strategies, when combined with carefully designed self-regular kernels, can achieve significant practical performance gains while maintaining rigorous theoretical complexity guarantees. The observed empirical complexity of $O(m^{0.199})$—substantially better than the worst-case $O(\sqrt{m})$ bound established in Theorem 4.2—suggests that real-world optimization problems exhibit favorable structural properties that the framework successfully exploits through its unified parametric design.

Algorithmic robustness across problem classes. The adaptability of the barrier parameter update mechanism, guided by the proximity measure $\delta(v)$ (Definition 2.2) and controlled by the self-regularity conditions (Definition 3.1), provides robust convergence across diverse problem classes. The framework handles strictly convex quadratic programs, nonconvex fractional programming instances, and large-scale separable problems with equal efficiency, achieving consistent speedups of 2×–30× over IPOPT while maintaining superior numerical precision (Tables 3–4). This robustness is particularly valuable in engineering applications where problem structure may not be known a priori, and where both solution quality and numerical stability are critical.

Computational trade-offs and application-specific tuning. The empirical results validate the theoretical insights from Section 4 regarding the trade-off between aggressive barrier reduction and proximity control. High-order kernels (ψ_4 with $p = 5$) paired with aggressive adaptive strategies ($k = \lceil\log(i+1)\rceil$) maximize computational speed but sacrifice ultimate precision, while lower-order kernels (ψ_3 with $p = 1$) paired with moderate strategies ($k = \lfloor i/2 \rfloor$) achieve machine-precision accuracy at the cost of slightly increased iteration counts. This flexibility enables application-specific performance tuning based on whether speed or accuracy is prioritized.

6. Conclusions

This paper introduced a unified parametric framework for primal-dual interior-point methods based on a generalized class of self-regular kernel functions defined through constituent functions $h(t)$ and $k(t)$. This construction subsumes several prominent kernel families while enabling the systematic derivation of the structural properties—strict convexity, growth bounds, and self-regularity—required for polynomial-time convergence. We established that the framework maintains the optimal worst-case complexity of $O(\sqrt{m}\log(m/\varepsilon))$ for convex optimization, while introducing adaptive barrier reduction strategies $\mu_{i+1} = (1 - \theta)^{k_i}\mu_i$ with iteration-dependent exponents $k_i \in \{1, i, \lceil\log(i + 1)\rceil, \lfloor i/2\rfloor\}$. These strategies significantly enhance practical performance while preserving global convergence guarantees.

Extensive numerical evaluation on problems with up to 10,000 variables demonstrated speedups of up to 37× over IPOPT, with KKT residuals approaching machine precision (10^{-15}). The observed empirical iteration complexity of $O(m^{0.199})$—representing a 60% reduction in the scaling exponent compared to the theoretical $O(\sqrt{m})$ bound—effectively bridges the gap between conservative worst-case complexity analysis and practical efficiency. Beyond the convex setting, the robust performance on nonconvex benchmarks highlights the framework's versatility as a stable and efficient heuristic for complex nonlinear systems.

Future research directions include: developing problem-structure-aware heuristics for automatic kernel parameter selection, exploiting separable structures for parallel Newton system factorizations in ultra-large-scale problems, and integrating the framework with branch-and-bound strategies for mixed-integer nonlinear programming. Additionally, investigating average-case complexity analysis could provide theoretical justification for the observed $O(m^{0.199})$ scaling law. These results establish that carefully designed self-regular kernels, paired with adaptive reduction strategies, deliver both theoretical rigor and exceptional computational performance for large-scale nonlinear optimization.

Author contributions

Mounia Laouar: Conceptualization, Methodology, data curation, formal analysis, Writing—Original draft; Mahmoud Brahimi: Conceptualization, Methodology, formal analysis, Validation, Writing—Original draft; Raouf Ziadi: Visualization, formal analysis, Investigation; Mohammed A. Saleh: Software, Supervision, Resources, Writing—Review and editing; Abdulgader Z. Almaymuni: Project administration, Software, Resources, Funding acquisition, Writing—Review and editing; Benmessaoud Chahinez: Writing—Review and editing. All authors reviewed the results and approved the final version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for the financial support (QU-APC-2026).

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. Y. Q. Bai, C. Roos, M. El Ghami, A primal-dual interior-point method for linear optimization based on a new proximity function, *Optim. Method. Softw.*, **17** (2002), 985–1008. https://doi.org/10.1080/1055678021000090024
2. Y. Q. Bai, M. El Ghami, C. Roos, A new efficient large-update primal-dual interior-point method based on a finite barrier, *SIAM J. Optimiz.*, **13** (2002), 766–782. https://doi.org/10.1137/S1052623401398132
3. Y. Q. Bai, C. Roos, A polynomial-time algorithm for linear optimization based on a new simple kernel function, *Optim. Methods Softw.*, **18** (2003), 631–646. https://doi.org/10.1080/10556780310001639735
4. Y. Q. Bai, M. El Ghami, C. Roos, A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization, *SIAM J. Optim.*, **15** (2004), 101–128. https://doi.org/10.1137/S1052623403423114
5. B. Bounibane, El. A. Djeffal, Kernel function-based interior-point algorithms for linear optimisation, *Int. J. Math. Model. Numer. Optim.*, **9** (2019), 158–177. https://doi.org/10.1504/IJMMNO.2019.098785
6. S. Boyd, L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.
7. R. H. Byrd, M. E. Hribar, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, *Math. Program.*, **89** (2000), 149–185. https://doi.org/10.1007/PL00011391
8. X. Z. Cai, G. Q. Wang, M. El Ghami, Y. J. Yue, Complexity analysis of primal-dual interior-point methods for linear optimization based on a new parametric kernel function with a trigonometric barrier term, *Abstr. Appl. Anal.*, **2014** (2014), 710158. https://doi.org/10.1155/2014/710158
9. R. Chalekh, El. A. Djeffal, Complexity analysis of an interior-point algorithm for CQP based on a new parametric kernel function, *Stat. Opt. Inf. Comput.*, **12** (2024), 153–166. https://doi.org/10.19139/soic-2310-5070-1761
10. E. Djeffal, M. Laouar, A primal-dual interior-point method based on a new kernel function for linear complementarity problem, *Asian-Eur. J. Math.*, **12** (2019), 1950101. https://doi.org/10.1142/S1793557120500011
11. E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2002), 201–213. https://doi.org/10.1007/s101070100263

12. B. El-Sobky, G. Ashry, An interior-point trust-region algorithm to solve a nonlinear bilevel programming problem, *AIMS Math.*, **7** (2022), 5534–5562. https://doi.org/10.3934/math.2022307

13. M. Laouar, M. Brahimi, I. E. Lakhdari, Kernel function with BFGS quasi-Newton methods for solving nonlinear semi-definite problems, *J. Math. Comput. Sci.*, **33** (2023), 1–16. https://doi.org/10.22436/jmcs.033.01.01

14. S. H. Low, Convex relaxation of optimal power flow, Part I: Formulations and equivalence, *IEEE Trans. Control. Netw. Syst.*, **1** (2014), 15–27. https://doi.org/10.1109/TCNS.2014.2309732

15. S. Mehrotra, On the implementation of a primal-dual interior point method, *SIAM J. Optim.*, **2** (1992), 575–601. https://doi.org/10.1137/0802028

16. Y. Nesterov, A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, Society for industrial and applied mathematics, 1994.

17. J. Nocedal, S. J. Wright, *Numerical optimization*, New York: Springer, 2006. https://doi.org/10.1007/978-0-387-40065-5

18. J. Peng, C. Roos, T. Terlaky, Self-regular functions and new search directions for linear and semidefinite optimization, *Math. Program.*, **93** (2002), 129–171. https://doi.org/10.1007/s101070200296

19. J. Peng, C. Roos, T. Terlaky, *Self-regularity: A new paradigm for primal-dual interior-point algorithms*, Princeton University Press, 2009.

20. C. Roos, A full-Newton step O(n) infeasible interior-point algorithm for linear optimization, *SIAM J. Optim.*, **16** (2006), 1110–1136. https://doi.org/10.1137/050623917

21. A. Q. Tian, F. F. Liu, H. X. Lv, Snow Geese algorithm: A novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems, *Appl. Math. Model.*, **126** (2024), 327–347. https://doi.org/10.1016/j.apm.2023.10.045

22. A. Q. Tian, J. S. Pan, H. X. Lv, Optimizing train scheduling in Heavy-Haul railways using diversified cooperative deep reinforcement learning, *Transp. Res. Rec.*, **2680** (2025), 286–312. https://doi.org/10.1177/03611981251364832

23. A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Program.*, **106** (2006), 25–57. https://doi.org/10.1007/s10107-004-0559-y

24. G. Wang, Y. Bai, Polynomial interior-point algorithm for $P_*(K)$ horizontal linear complementarity problem, *J. Comput. Appl. Math.*, **233** (2009), 248–263. https://doi.org/10.1016/j.cam.2009.07.014

25. S. J. Wright, *Primal-dual interior-point methods*, Society for Industrial and Applied Mathematics, 1997.

AIMS Press