Noname manuscript No.
(will be inserted by the editor)

An efficient penalty decomposition algorithm for minimization over sparse symmetric sets

Ahmad Mousavi · Morteza Kimiaei* · Saman Babaie–Kafaki · Vyacheslav Kungurtsev

the date of receipt and acceptance should be inserted later

Abstract This paper proposes an improved quasi-Newton penalty decomposition algorithm for the minimization of continuously differentiable functions, possibly nonconvex, over sparse symmetric sets. The method solves a sequence of penalty subproblems approximately via a two-block decomposition scheme: the first subproblem admits a closed-form solution without sparsity constraints, while the second subproblem is handled through an efficient sparse projection over the symmetric feasible set. Under a new assumption on the gradient of the objective function, weaker than global Lipschitz continuity from the origin, we establish that accumulation points of the outer iterates are basic feasible and cardinality-constrained Mordukhovich stationarity points. To ensure robustness and efficiency in finite-precision arithmetic, the algorithm incorporates several practical enhancements, including an enhanced line search strategy based on either backtracking or extrapolation, and four inexpensive diagonal Hessian approximations derived from differences of previous iterates and gradients or from eigenvalue-distribution information. Numerical experiments on a diverse benchmark of 30 synthetic and data-driven test problems,

A. Mousavi
Department of Mathematics and Statistics, American University, Washington, DC, USA
E-mail: mousavi@american.edu

M. Kimiaei (corresponding author)
Fakultät für Mathematik, Universität Wien, Oskar-Morgenstern-Platz 1, A-1090, Wien, Austria
E-mail: kimiaeim83@univie.ac.at

S. Babaie–Kafaki
Faculty of Engineering, Free University of Bozen–Bolzano, Bolzano 39100, Italy
E-mail: saman.babaiekafaki@unibz.it

V. Kungurtsev
Department of Computer Science, Czech Technical University, Karlovo Namesti 13, 121 35 Prague 2, Czech Republic
E-mail: vyacheslav.kungurtsev@fel.cvut.cz

including machine-learning datasets from the UCI repository and sparse symmetric instances with dimensions ranging from 10 to 500, demonstrate that the proposed algorithm is competitive with several state-of-the-art methods in terms of efficiency, robustness, and strong stationarity.

Keywords Sparse optimization · penalty decomposition method · diagonal quasi-Newton method · line search method · global convergence

2000 AMS Subject Classification: 90C26; 90C30; 65K05.

1 Introduction

In modern applications such as machine learning, signal processing, and data mining, high-dimensional data pose significant computational and modeling challenges. To address these challenges, one widely adopted strategy is to impose **sparsity**, i.e., to focus on a limited number of relevant features while discarding the rest. Sparsity not only reduces computational cost but also improves interpretability, making it a key principle in large-scale optimization.

A central challenge in sparse optimization is understanding the structure of the feasible set. Many important problem classes are defined over **symmetric sets**, i.e., sets that are invariant under permutations or sign changes of the variables. Examples include the full space, the nonnegative orthant, the simplex, norm balls, and box constraints. Each of these sets induces a specific structure on the feasible solutions, which must be carefully respected in sparse projection algorithms.

Minimization over sparse symmetric sets is challenging because it requires combining the combinatorial nature of sparsity with structural constraints. Efficient projection rules exploit the geometry of each set: in the full space, the projection keeps the largest components; in the orthant, nonnegativity must be preserved; in the simplex or unit-sum set, normalization is enforced; and for norm or box constraints, rescaling or clipping may be needed. Understanding these structural nuances is crucial for designing algorithms that are both computationally efficient and theoretically sound.

1.1 Problem Definition

Following recent research trends, optimization problems that combine sparsity with additional structural constraints have gained considerable attention. Motivated by applications in signal recovery, image processing, and data compression, we study the following general cardinality-constrained optimization

problem:

$$\min_{x \in C \cap C_s} f(x), \tag{CCOP}$$

where the objective function $f : C \cap C_s \to \mathbb{R}$ is assumed to be twice continuously differentiable, with gradient $g(x) := \nabla f(x)$, but not necessarily convex, and

$$C_s := \{x \in \mathbb{R}^n \mid \|x\|_0 \leq s\}, \tag{1}$$

and $C \subseteq \mathbb{R}^n$ is a closed and convex set representing additional structure. Examples of symmetric sets include $\mathbb{R}^n$, the nonnegative orthant, the simplex, ℓ_p-norm balls, and box constraints. These sets are invariant under sign changes and/or permutations of coordinates, a property that strongly influences sparse projection rules.

The set C_s enforces the **sparsity constraint**, with $s \in \mathbb{Z}_+$ denoting the target sparsity level ($s < n$), and $\|x\|_0$ indicating the number of nonzero components of x. Despite its simple definition, C_s is highly nonconvex and disconnected, and problems of the form (CCOP) are in general NP-hard. It is well known that even testing the feasibility of the sparsity set is NP-complete [19,22], so that (CCOP) inherits this fundamental computational hardness.

In the context of such problems, several stationarity notions have been proposed, including Lu–Zhang points [23], basic feasible points [3], L-stationarity [3], and Mordukhovich-stationarity (M-stationarity) points [19,26,33]. These concepts play a key role in analyzing the convergence behavior of penalty decomposition and related algorithms, and will be revisited later in the paper.

This broad formulation covers many important models. For instance, when f is quadratic and $C = \mathbb{R}^n$, one recovers the classical sparse recovery problem in compressive sensing. Incorporating additional convex sets C, such as the nonnegative orthant, the simplex, or norm constraints, yields a wide class of structured sparse optimization problems. In the terminology of [19,33], problem (CCOP) belongs to the class of mathematical programs with cardinality constraints.

1.2 Related Work

Sparse optimization problems aim to find solutions with few nonzero components, often under structural or functional constraints. A variety of algorithmic approaches have been proposed to tackle these problems, each with its own advantages and limitations. In this section, we categorize the most prominent methods and discuss their similarities, differences, strengths, and weaknesses, with a particular focus on how penalty decomposition methods compared to

the others. We begin with approaches offering the weakest theoretical guarantees and proceed toward increasingly stronger ones.

Greedy algorithms such as orthogonal matching pursuit and forward/backward selection incrementally construct sparse solutions by selecting variables according to local criteria. These methods are computationally efficient and easy to implement. Yet, they often suffer from suboptimal selections and are sensitive to noise and variable correlations, which limit their robustness and ability to scale reliably in high-dimensional, correlated settings [13, 38].

A second class providing relatively weak guarantees is **convex relaxation**, where the original nonconvex sparsity constraint (typically represented by the ℓ_0-norm) is replaced with a convex surrogate like the ℓ_1-norm. Popular methods in this class include Basis Pursuit, `LASSO`, and the Elastic Net [12,16,35]. These approaches benefit from convex optimization theory and mature solvers, but they typically return only approximately sparse solutions and may introduce significant shrinkage bias.

Moving toward stronger guarantees, **thresholding and iterative shrinkage methods**, including Iterative Hard Thresholding (`IHT`) and proximal gradient variants, project intermediate solutions onto the set of s-sparse vectors [8]. These methods are well-suited for large-scale problems and admit global convergence guarantees under restricted conditions. Nonetheless, their performance is sensitive to step-size rules, tuning parameters, and problem conditioning [6,17,40].

Stationarity-based algorithms form a considerably stronger class. Multiple frameworks have been designed to compute stationarity points of nonconvex sparse optimization problems [3,4,22,23]. These methods aim to satisfy first-order necessary conditions—even in nonsmooth and nonconvex settings—but often converge only to generalized notions such as Lu–Zhang stationarity points, which may still fail to guarantee strict feasibility or desirable structural properties [22, Example 2.1].

In contrast, **mixed-integer and combinatorial methods** directly encode sparsity through binary variables, enabling them to certify global optimality. Examples include mixed-integer quadratic programming formulations [7,10]. Their major limitation is scalability, as the combinatorial search space grows exponentially with problem size.

Among these diverse methodological families, we emphasize **penalty decomposition (`PD`) methods** because they serve as the foundation of the algorithms analyzed in this paper and offer a balanced trade-off between theoretical structure and computational practicality. Penalty and augmented Lagrangian techniques incorporate sparsity constraints through penalization or variable decompositions, enabling the separation of difficult nonsmooth or combinatorial components from smooth differentiable ones. `PD` methods have received

increasing attention due to their ability to decouple complex constraints and nonconvex cost functions into tractable subproblems solvable via block coordinate descent [23]. They are often less sensitive to initializations, especially when warm-started using simpler heuristics.

Furthermore, PD algorithms are compatible with limited-memory quasi-Newton updates [11,20], providing scalability and effective use of curvature information while allowing inexact line searches [30]. The flexibility of PD frameworks is highlighted by numerous applications: Dong and Zhu [15] integrated IHT-type updates for adaptive sparsity-level detection; Lu et al. [24] applied PD to rank minimization; and Kanzow and Lapucci [18] proposed an inexact PD method for geometric constraints such as cardinality constraints and rank constraints. Additional applications span multi-objective sparse optimization [21], wavelet-frame ℓ_0 image reconstruction [39], sparse time-series filtering [32], cardinality-constrained portfolio optimization [29], and nuclear norm minimization with ℓ_1 fidelity terms [37]. Algorithmic relaxations designed to reduce subproblem complexity have also been proposed in [34].

Compared to the other paradigms discussed above, PD methods preserve the problem's inherent structure while enabling effective enforcement of sparsity and feasibility. They exhibit robustness and flexibility, particularly in extensions involving structured sparse sets (e.g., simplex constraints and mixed-norm balls). When paired with quasi-Newton updates, they offer both scalability and accurate practical performance. Thus, while simpler methods may provide speed or convex surrogates ensure elegant theory, PD represents a compelling middle ground.

However, existing inexact PD algorithms (e.g., [22,23]) are typically guaranteed to converge only to Lu–Zhang stationarity points and, under mild assumptions, to BF points. These concepts remain weaker than cardinality-constrained Mordukhovich (CC-M) stationarity [19], which is the strongest *variationally necessary* first-order optimality condition for cardinality-constrained optimization in the absence of constraint qualifications. Stronger notions, such as CC-S (strong stationarity), may exist but are not guaranteed to hold at all local minimizers and typically require additional regularity or support-identification assumptions. Developing PD variants that converge directly to CC-M-stationarity points would bridge this gap between algorithmic guarantees and advanced optimality theory, thereby significantly enhancing the robustness and applicability of PD methods across broader classes of nonsmooth and geometrically constrained optimization problems [18,19,33].

1.3 Main Contributions of our Work

In this study, we propose an improved quasi-Newton penalty decomposition algorithm, called PD-QN, for solving optimization problems involving contin-

uously differentiable functions over sparse symmetric sets. Like the classical penalty decomposition algorithm [22,23], PD-QN approximates the solution of a sequence of penalized subproblems using a two-block decomposition scheme. At each iteration of its inner loop, PD-QN solves the first subproblem, denoted by (P_x), in closed form with respect to the variable x, without sparse symmetric sets. It then solves the second subproblem, (P_y), with respect to y **restricted to its current support**. This restricted minimization is performed explicitly and at low cost, and—especially—it introduces a new feature that was not previously incorporated into existing PD algorithms. By solving over the current support, PD-QN both preserves sparsity and significantly improves computational efficiency over prior methods.

Current inexact PD algorithms guarantee convergence to Lu–Zhang stationarity points and, under mild assumptions, to basic feasible (BF) points. These stationarity concepts, however, are tailored to purely cardinality-constrained or symmetric-set formulations and therefore do not fully capture the broader geometry arising when cardinality constraints are coupled with general inequality constraints. In contrast, CC-M-stationarity—the appropriate Mordukhovich-type stationarity notion for fully general cardinality-constrained problems—is strictly stronger and provides a unifying optimality concept beyond the symmetric-set setting.

Developing PD variants that converge directly to CC-M-stationarity points—rather than merely to Lu–Zhang or BF points—thus bridges a substantial gap between existing algorithmic guarantees and modern variational optimality theory for general cardinality-constrained programs. While convergence to CC-S-stationarity points cannot be expected in general without additional assumptions, CC-M represents the strongest stationarity notion that can be guaranteed globally for penalty decomposition methods (see, e.g., [18,19]).

1.3.1 Algorithmic Features of our Methodology

Our main algorithmic features are summarized as follows:

(i) **A new reformulation of the two penalty subproblems in the inner loop:** In the proposed formulation, the subproblem (P_x) is entirely unconstrained—there are no sparse symmetric sets. In contrast, the subproblem (P_y) is solved over a sparse symmetric set. In particular, (P_y) is minimized **only over the current support** of the iterate, rather than over the full space. This support-aware minimization is a key novelty of PD-QN, distinguishing it from existing penalty decomposition algorithms. The motivation for this reformulation stems from the observation that alternating between two fully constrained subspaces, as done in classical PD methods, can be computationally inefficient. In practice, solving (P_x) without sparse symmetric sets is significantly simpler, as it admits a closed-form

solution. Meanwhile, the restricted form of (P_y), despite enforcing sparsity and symmetry, remains tractable and can be solved explicitly using a low-cost strategy. Beyond its computational benefits, this reformulation is also critical for the convergence analysis: by restricting (P_y) to the current support, `PD-QN` ensures that its iterates satisfy a basic feasibility condition, and ultimately converge to a `BF` point of the original problem.

(*ii*) **Construction of an accelerated line search method to efficiently solve** (P_x)**:** Our line search is performed either by a backtracking or an extrapolation framework, starting with the unit step size, which is classically advisable for the quasi-Newton algorithms, especially near the optimal solution. If a reduction in the model function value is found with the initial setting $\alpha = 1$, then extrapolation is performed to leave the regions containing a saddle point or a maximizer. Otherwise, the step size is reduced as long as the line search condition is violated. In this strategy, only one objective function evaluation is required at the accepted point. In contrast, the model function values at the other trial points are computed without any additional objective function evaluations. Hence, our line search has a lower computational cost than any inexact line search that computes the objective function value at each trial point of the line search procedure.

(*iii*) **Construction of four diagonal Hessian approximations to handle large-scale problems:** Three of such diagonal formulas are constructed based on the classic limited-memory `BFGS` formula by forming and updating two matrices whose columns are the most recent step change and the most recent gradient change. The other approximation is devised based on improving the distribution of the diagonal entries (or equivalently, the eigenvalues) of the diagonal Hessian estimation, as a measure to promote well-conditioning. Since, unlike the `BFGS` update, these four diagonal Hessian approximations do not necessarily guarantee the curvature condition [31], some proper safeguards are considered for the given Hessian approximations as well.

(*iv*) **Warm-start and stagnation recovery:** We begin with a warm-start phase using the `BFS` (the basic feasible search of [4, Algorithm 5]) routine tailored to sparse structures, which quickly proposes a promising support and refines it with a short restricted `FISTA` [5] update. This produces a strong initial point and significantly reduces the effort required by the main solver. If progress later completely halts, we apply the lightweight `PSS` (the sparse-simplex method of [3]) perturbation, which performs small support-growth or swap moves combined with simple coordinate corrections. This mechanism provides sufficient variation in the support to escape poor stationarity points and enables the main algorithm to resume stable convergence.

1.3.2 Strong Global Convergence Feature

It is well understood that the convergence analysis of many classical optimization algorithms often relies on assumptions regarding the regularity of the objective function or its gradient. A common and powerful assumption is the **Lipschitz continuity of the gradient**, which ensures that the function behaves in a sufficiently smooth way so that its values can be well approximated by a quadratic model based on local gradient information. In practical terms, this assumption prevents the function from changing too abruptly, which is critical for establishing convergence rates of gradient-based algorithms.

A related but weaker notion is sometimes referred to as **Lipschitz continuity from the origin**. This condition requires only that the gradient grows at most linearly with the distance from the origin. Unlike full Lipschitz continuity, however, it does not provide uniform control over the gradient across the entire domain and thus offers a much weaker form of regularity.

These two assumptions serve different analytical purposes and should not be confused. Standard Lipschitz continuity of the gradient is a strong smoothness condition that underpins many classical theoretical guarantees, whereas Lipschitz continuity from the origin is merely a basic growth condition that conveys far less information about the function's behavior. Confusing the two may lead to oversimplified or even incorrect conclusions in theoretical developments.

In the literature on `PD` methods, convergence has typically been established under relatively strong smoothness assumptions. For example, the original analysis by Lu and Zhang [23] required Lipschitz continuity of the gradient, while the more recent inexact `PD` framework of Lapucci et al. [22] still relied on comparable regularity conditions to guarantee convergence to Lu–Zhang stationarity points. In both cases, the analysis crucially depends on global gradient smoothness or growth conditions.

In this work, we establish global convergence of our algorithm under a new and even milder assumption than Lipschitz continuity from the origin. Unlike classical assumptions that require either full gradient smoothness or uniform growth bounds, our analysis only relies on a relaxed gradient growth condition, which allows the gradient to grow linearly outside a bounded region. Crucially, convergence to a `BF` point is guaranteed not only by this weaker assumption but also by a distinctive feature of our algorithm: in each iteration, the subproblem (P_y) is minimized over the current support of the iterate. This support-restricted formulation plays a central role in our analysis and, to the best of our knowledge, yields the first convergence guarantee for penalty decomposition methods that ensures convergence specifically to `BF` and `CC-M`-stationarity points.

In addition to these relaxed smoothness requirements, our convergence analysis exploits a key structural property of the penalized models, namely their

uniform strong convexity under bounded penalty parameters. Owing to the quadratic form of the penalty models and the uniform positive definiteness of their Hessians, the models satisfy global quadratic growth and error bound conditions with constants that are independent of the outer iteration index and the penalty parameter. These properties yield a finite-length argument for the outer iterates and ensure convergence of the **entire** sequence, rather than merely subsequential convergence. Although this behavior can be interpreted within the Kurdyka–Łojasiewicz (KL) framework (with exponent $\frac{1}{2}$), no explicit KL assumption is required here: the result follows directly from strong convexity and descent. To the best of our knowledge, leveraging this uniform strong convexity to obtain full-sequence convergence is not standard in existing penalty decomposition analyses and provides an additional layer of robustness in our global convergence guarantees.

Finally, we emphasize that our convergence results are conceptually similar to those established for general nonsmooth or geometrically constrained settings [19], in which penalty decomposition schemes also guarantee convergence to `CC-M`-stationarity points. The key difference is that our analysis is tailored to cardinality-constrained optimization and exploits the support-restricted subproblem structure. In contrast, the existing results address other classes of nonsmooth feasibility sets. Thus, our contribution complements the general theory by providing the first `BF`/`CC-M`-stationarity convergence guarantees for this particular but practically important problem class.

1.3.3 Our Computational Plans

We perform numerical experiments on a benchmark set of 30 test problems, including the datasets Iris, Wine, and Boston Housing (from the UCL repository), as well as several sparse symmetric instances discussed in [4], together with sparsity-constrained examples drawn from the survey article [36]. The problem dimensions in our test suite range from 10 to 500.

We compare our method against several state-of-the-art algorithms—iterative hard thresholding [3], the sparse simplex method [3], greedy sparse simplex [3], basic feasible search [4], and zero-`CW` search [4]—which are commonly used to compute approximate global minimizers and stationarity points for cardinality-constrained problems. The selected test problems are deliberately challenging, combining explicit cardinality constraints, medium- to high-sparsity regimes, and symmetric feasible sets that give rise to multiple competing stationarity supports. All methods are evaluated using a unified stopping framework based on objective reduction and violations of `CC-S` (strong) stationarity, as detailed in Section 5.1. The use of `CC-S` in the stopping criteria serves purely as a numerical quality measure: since `CC-S` implies both `CC-M` and `BF` stationarity, any iterate satisfying the numerical stopping conditions necessarily exceeds the theoretical guarantees required by our convergence analysis.

The results demonstrate that our algorithm is competitive with these established techniques.

1.4 Paper Organization

The organization of our study is outlined as follows. Section 2 is devoted to a detailed discussion of foundational concepts and general methodological tools used throughout the paper. Section 3 introduces our new algorithm within the quasi-Newton penalty decomposition framework, together with its main computational components. Convergence properties of the proposed method are established in Section 4. Section 5 reports extensive numerical experiments illustrating the practical efficiency of the algorithm. Finally, concluding remarks are presented in Section 6.

1.5 Supplemental Theory and Algorithmic Components

Additional results and algorithmic details are provided in the supplementary material available at [28], organized as follows: Section 2 lists explicit stationarity conditions for several convex sets that appear in our analysis. Section 3 reviews basic feasibility and L-stationarity under symmetry assumptions and includes a practical test for an approximate notion of basic feasibility based on a single super support set. Section 4 contains the proof of Lemma 1, which establishes the cone continuity property for convex symmetric sets. Section 5 presents the complete proof of Theorem 1 using standard calculus rules for Fréchet normal cones. Section 6 compares basic feasibility with the various CC-stationarity notions and clarifies their position within the stationarity hierarchy. Section 7 describes practical enhancements of our method, including an improved line search and several diagonal Hessian approximations. Section 8 summarizes sparse projection algorithms for symmetric sets, and Section 9 reports additional numerical comparisons among our algorithm variants.

An earlier unpublished version of this work appears in [27]. The present paper differs substantially in that we now establish convergence to an M-stationarity point, whereas the preliminary version only proved convergence to a Lu–Zhang stationarity point. In addition, the algorithm has been extended from handling cardinality problems to addressing the full class of cardinality-constrained optimization problems over symmetric sets, which includes the bound-constrained case. As a result, the current version offers a stronger theoretical foundation together with markedly improved numerical performance.

2 Preliminaries and Methodological Foundations

This section begins by outlining the key foundational concepts that underpin our analysis and approach. We then sequentially introduce the notions of symmetric sets, sparse projection techniques tailored to these sets, and various first-order optimality conditions associated with the corresponding optimization problems. Most of the definitions presented here are drawn from the work of Beck and Hallak [4], Kanzow et al. [19], and Mordukhovich [25]. We include them to ensure the paper is self-contained and accessible, allowing readers to follow the developments without needing to consult the original reference [4, 19, 25].

2.1 Notation and Foundational Concepts

Let $[n] := \{1, 2, \ldots, n\}$. The n-dimensional simplex is

$$\Delta_n := \left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = 1,\ x_i \geq 0\ \forall i \in [n] \right\},$$

i.e., the set of all nonnegative vectors with components summing to one. The unit-sum set is

$$\Delta'_n := \left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = 1 \right\},$$

which imposes the same equality constraint but without sign restrictions. The sign vector $\operatorname{sign}(x)$ of a given $x \in \mathbb{R}^n$ is the vector whose ith component is $\operatorname{sign}(x)_i := 1$ if $x_i \geq 0$ and $\operatorname{sign}(x)_i := -1$ otherwise. Given a set $S \subseteq \mathbb{R}^n$ and a vector $x \in \mathbb{R}^n$, the orthogonal projection of x onto S is defined by

$$P_S(x) = \operatorname{argmin} \{\|y - x\| : y \in S\},$$

where $\|.\|$ denotes the ℓ_2-norm on $\mathbb{R}^n$. If the set S is closed, then $P_S(x)$ is nonempty. Furthermore, if S is also convex, then $P_S(x)$ is a singleton, and we identify $P_S(x)$ with the unique vector that it contains.

Given a closed and convex set $C \subseteq \mathbb{R}^n$, and a vector $x \in \mathbb{R}^n$, the sparse projection problem seeks to find an element in the orthogonal projection set of x onto $C \cap C_{\mathrm{s}}$, where C_{s} is the set of all vectors in $\mathbb{R}^n$ with at most s nonzero components. Formally, the sparse projection set is defined by

$$P_{C_{\mathrm{s}} \cap C}(x) = \operatorname*{argmin}_{z \in C \cap C_{\mathrm{s}}} \|z - x\|^2.$$

We refer to $P_{C_{\mathrm{s}} \cap C}$ as the s-sparse projection set onto C, and any element of this set is called an s-sparse projection vector onto C, or simply a sparse projection

vector. Since the intersection $C \cap C_{\mathrm{s}}$ is closed, the set $P_{C_{\mathrm{s}} \cap C}(x)$ is nonempty for any $x \in \mathbb{R}^n$. However, because $C_{\mathrm{s}} \cap C$ is nonconvex, the projection set $P_{C_{\mathrm{s}} \cap C}(x)$ is not necessarily a singleton. When $C = \mathbb{R}^n$, the sparse projection reduces to $P_{C_{\mathrm{s}} \cap \mathbb{R}^n}(x) = P_{C_{\mathrm{s}}}(x)$, which consists of all vectors formed by retaining the s components of x with the largest absolute values (setting all others to zero). If there are some ties among the largest absolute values, multiple selections are possible, and hence, the projection set can contain more than one vector. For further details, refer to [28, Section 2], which is based on the work presented in [4].

For any $p \geq 1$, the ℓ_p-ball in the space $\mathbb{R}^n$, centered at the origin with radius 1, is defined as

$$B_p^n[0,1] = \{x \in \mathbb{R}^n \mid \|x\|_p \leq 1\},$$

where $\|x\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ is the ℓ_p-norm of x.

The **support set** $I_1(x) := \{i \in [n] \mid x_i \neq 0\}$ of a vector $x \in \mathbb{R}^n$ and its complement, called the **off-support set** of a vector $x \in \mathbb{R}^n$, $I_0(x) := \{i \in [n] \mid x_i = 0\}$ are also defined accordingly. A vector $x \in \mathbb{R}^n$ has a **full support** if $\|x\|_0 = s$, and an **incomplete support** if $\|x\|_0 < s$. A set $\mathcal{L} \subseteq [n]$ is called a **super support** of a vector $y \in C_{\mathrm{s}} \cap C$ if $I_1(y) \subseteq \mathcal{L}$ and $|\mathcal{L}| = s$. Note that if y has full support, the only super support set is the support set itself. Otherwise, the number of possible super supports is

$$\binom{n - \|y\|_0}{s - \|y\|_0}.$$

Given a vector $x \in \mathbb{R}^n$, and a subset of indices $\mathcal{L} \subseteq [n]$, $x_{\mathcal{L}} \in \mathbb{R}^{|\mathcal{L}|}$ denotes the vector composed of the components of x indexed by $\mathcal{L}$. Let $U_{\mathcal{L}}$ denote the submatrix of the $n \times n$ identity matrix I_n formed by selecting the columns corresponding to the index set $\mathcal{L}$; then $x_{\mathcal{L}} = U_{\mathcal{L}}^T x$. Moreover, if $\mathcal{L}$ is a super support of a vector $x \in \mathbb{R}^n$, then $x = U_{\mathcal{L}} x_{\mathcal{L}}$. Given a set $C \subseteq \mathbb{R}^n$, the restriction of C to the index set $\mathcal{L}$ is defined as

$$C_{\mathcal{L}} := \left\{x \in \mathbb{R}^{|\mathcal{L}|} : U_{\mathcal{L}} x \in C\right\}.$$

Given a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ and a subset $\mathcal{L} \subseteq [n]$, the restriction of the gradient $g(x)$ to the index set $\mathcal{L}$ is denoted by $g_{\mathcal{L}}(x) = U_{\mathcal{L}}^T g(x)$.

2.2 Symmetric Sets

Let $\mathfrak{S}_n$ denote the **symmetric group of all permutations** of the set of indices $[n]$. For a vector $x \in \mathbb{R}^n$ and a permutation $\pi \in \mathfrak{S}_n$, the permuted

vector $x^\pi \in \mathbb{R}^n$ is defined component-wise as $(x^\pi)_i := x_{\pi(i)}$. For example, let

$$x = \begin{pmatrix} 4 \\ 1 \\ 6 \end{pmatrix} \in \mathbb{R}^3, \quad \pi \in \mathfrak{S}_3 \ \text{ with } \pi(1) = 3,\ \pi(2) = 2,\ \pi(3) = 1.$$

Then, the permuted vector is

$$x^\pi = \begin{pmatrix} x_{\pi(1)} \\ x_{\pi(2)} \\ x_{\pi(3)} \end{pmatrix} = \begin{pmatrix} x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} 6 \\ 1 \\ 4 \end{pmatrix}.$$

A permutation $\pi \in \tilde{\mathfrak{S}}_n$ is called a **sorting permutation** of a vector $x \in \mathbb{R}^n$ whose entries are rearranged in a non-increasing order in the sense that $x_{\pi(1)} \geq x_{\pi(2)} \geq \cdots \geq x_{\pi(n)}$. Here, $\tilde{\mathfrak{S}}_n$ denotes the sorting permutation group over the set of indices $[n]$. For any permutation $\pi \in \tilde{\mathfrak{S}}_n$, we define

$$S^\pi_{[j_1,j_2]} = \begin{cases} \{\pi(j_1), \pi(j_1+1), \ldots, \pi(j_2)\}, & \text{if } 0 < j_1 \leq j_2 \leq n, \\ \emptyset, & \text{otherwise.} \end{cases} \tag{2}$$

Let $C \subseteq \mathbb{R}^n$ be a closed and convex set; then, C is called
• **type-1 symmetric**, if for any $x \in C$ and any permutation $\pi \in \mathfrak{S}_n$, $x^\pi \in C$;
• **nonnegative**, if for every $x \in C$, $x_i \geq 0$ for all i;
• **type-2 symmetric set**, if it is a type-1 symmetric set and if for any $x \in C$, any $\pi \in \mathfrak{S}_n$, and any $y \in \{-1, 1\}^n$, $x \circ y \in C$, with $(x \circ y)_i = x_i y_i$ for all $i \in [n]$.

Let $C \subseteq \mathbb{R}^n$ be a closed and convex set that is either a nonnegative type-1 symmetric set or a type-2 symmetric set. Let $x \in \mathbb{R}^n$, and $\pi \in \tilde{\mathfrak{S}}(p(x))$, where the symmetry function $p : \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$p(x) = \begin{cases} x, & \text{if } C \text{ is a nonnegative type-1 symmetric set,} \\ |x|, & \text{if } C \text{ is a type-2 symmetric set.} \end{cases} \tag{3}$$

Here, $|x|$ denotes the component-wise absolute value of x (this function is used to define a common sorting permutation $\pi \in \tilde{\mathfrak{S}}(p(x))$ for both cases).

The above definitions, $p(x)$, $\tilde{\mathfrak{S}}$, and $S^\pi_{[j_1,j_2]}$, will be used in lines 9 and 10 of our new algorithm (Algorithm 1, below).

Symmetric sets of type-1 and type-2 frequently arise as feasible regions in optimization problems. The entire space $\mathbb{R}^n$ is both a type-1 and type-2 symmetric set. The nonnegative orthant $\mathbb{R}^n_+$ is a type-1 set and, more specifically, also a nonnegative type-1 set, but not type-2. The unit simplex Δ_n shares the same properties as the nonnegative orthant—it is type-1 and nonnegative type-1,

but not type-2. The unit sum set Δ_n' is only a type-1 set. The ℓ_p-ball $B_p^n[0,1]$ (for $p \geq 1$) is both a type-1 and type-2 symmetric set. Lastly, the box constraints set $[\ell, u]^n$, with $\ell < u$, is type-1 but neither nonnegative type-1 nor type-2.

2.3 Optimality Conditions

This section presents an overview of the first-order optimality conditions for smooth optimization problems over closed and convex sets. We begin by reviewing classic stationarity conditions within the framework of convex analysis and then extend the discussion to the sparse optimization problem (CCOP), which involves the intersection of a symmetric constraint set C and the nonconvex sparsity set C_{s}. By examining the structure of this composite feasible region, we introduce existing stationarity concepts that are well-suited for nonconvex problems with embedded sparsity constraints. These conditions form the theoretical foundation for the development and analysis of the proposed algorithm. For further details on the optimality conditions, see [28, Section 3].

2.3.1 For Smooth Problems over Convex Sets

We consider the convexly constrained optimization problem

$$\min\{f(x) : x \in C\},$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $C \subseteq \mathbb{R}^n$ is a nonempty closed convex set. A vector $x^* \in C$ is called a **stationarity point** of this problem if

$$g(x^*)^T(x - x^*) \geq 0, \qquad \forall x \in C,$$

where $g(x^*) := \nabla f(x^*)$.

This variational inequality expresses that there are no feasible descent directions at x^*. Equivalently, it can be written as the fixed-point condition

$$x^* = P_C\big(x^* - \tfrac{1}{L}\, g(x^*)\big)\,, \qquad \text{for some } L > 0,$$

where $P_C(\cdot)$ denotes the Euclidean projection onto C. See [28, Remark 2.1] for the explicit stationarity conditions in [28, Section 3].

2.3.2 For Problem (CCOP)

Basic Feasible Points and Optimality: A vector $x \in C_s \cap C$ is called a **basic feasible (BF) point** of problem (CCOP) if, for any super support set $\mathcal{L}$ of x, there exists a scalar $L > 0$ such that

$$x_{\mathcal{L}} = P_{C_{\mathcal{L}}}\left(x_{\mathcal{L}} - \frac{1}{L} g_{\mathcal{L}}(x)\right). \tag{4}$$

If $|I_1(x)| = s$, then the only super support set is the support itself, and hence, the BF condition reduces to

$$x_{I_1(x)} = P_{C_{I_1(x)}}\left(x_{I_1(x)} - \frac{1}{L} g_{I_1(x)}(x)\right), \tag{5}$$

which reduces to the standard projection condition, which is necessarily satisfied at BF points with full support. However, it is not sufficient when the support is incomplete. The BF condition is equivalent to requiring that, for any super support set $\mathcal{L}$ of x, the vector $x_{\mathcal{L}}$ is a stationarity point of the following convex-constrained optimization problem:

$$\min\{f(U_{\mathcal{L}} w) : w \in C_{\mathcal{L}}\}.$$

Although condition (4) is written using L, it is essentially independent of the choice of L, and can alternatively be expressed as

$$g_{\mathcal{L}}(x)^T (y_{\mathcal{L}} - x_{\mathcal{L}}) \geq 0, \qquad \text{for all } y \in C \text{ with } I_1(y) \subseteq \mathcal{L}.$$

Let x^* be an optimal solution of the problem (CCOP). Then, it has been shown in [4, Theorem 5.1] that x^* is a BF point of (CCOP).

It is important to note that when the support of a vector x is not full, verifying whether x is a BF point, in principle, requires checking condition (4) for each possible choice of a super support set. The number of such checks is

$$\binom{n - \|x\|_0}{s - \|x\|_0}.$$

However, when the set C is either a nonnegative type-1 symmetric set or a type-2 symmetric set, there exist simpler procedures for verifying basic feasibility in the case of incomplete support. Specifically, it is sufficient to verify that condition (4) holds for a particular super support set.

When x^* has full support (i.e., $\|x^*\|_0 = s$), the BF condition reduces to the standard first-order rules for the underlying convex set C, restricted to the active support $I_1(x^*)$. In other words, the explicit formulas given in [28, Remark 2.1] for smooth convex problems remain valid, but they apply only to the indices in $I_1(x^*)$.

By contrast, if x^* has incomplete support ($\|x^*\|_0 < s$), the BF condition requires projected-gradient stationarity on every super-support $\mathcal{L}$, i.e., with respect to the convex restriction $C_{\mathcal{L}}$ rather than the entire set C.

As previously mentioned, the concept of BF is tied to stationarity with respect to the restriction of the set C to super support sets of the vector. However, it provides no guarantee regarding the optimality of the support itself. In this sense, the BF condition is relatively weak. We now introduce a stronger notion known as the L-stationarity condition.

***L*-Stationarity.** For a constant $L > 0$, a vector $x \in C_s \cap C$ is said to be an L-stationarity point of problem (CCOP) if

$$x \in P_{C_s \cap C}\left(x - \frac{1}{L} g(x)\right).$$

This condition implies that x is a fixed point of the projected-gradient step with the step size $1/L$, over the nonconvex feasible set $C_s \cap C$. Let $x^* \in C_s \cap C$ be an L-stationarity point of (CCOP). Then, as shown in [4, Lemma 5.2], x^* is a BF point of (CCOP).

Lu–Zhang stationarity Point: Lu and Zhang [23] defined another optimality condition for the problem CCOP. The vector $x \in \mathbb{R}^n$ is called a Lu–Zhang stationarity point if and only if there exists an index set $\mathcal{L} \subseteq [n]$ with $|\mathcal{L}| = s$ such that

$$\begin{cases} g_i(x) = 0, & \text{for all } i \in \mathcal{L} \subseteq [n] \text{ with } |\mathcal{L}| = s, \\ x_i = 0, & \text{for all } i \in \mathcal{L}^c, \end{cases} \tag{6}$$

where

$$\mathcal{L}^c := [n] \setminus \mathcal{L}. \tag{7}$$

They also showed that when $C = \mathbb{R}^n$, any optimal solution of problem (CCOP) is a Lu–Zhang stationarity point for the same problem. Moreover, when $C = \mathbb{R}^n$, any BF point of problem (CCOP) is also a Lu–Zhang stationarity point; however, for general closed convex C this implication may fail—see [22, Example 2.1].

Let $S \subseteq \mathbb{R}^n$ be a closed set and $\bar{x} \in S$. To rigorously characterize the stationarity of problem (CCOP), we recall the standard normal cones from variational analysis:

- The **Fréchet (regular) normal cone** is defined as:

$$N_S^F(\bar{x}) := \left\{ \gamma \in \mathbb{R}^n \mid \limsup_{x \xrightarrow{S} \bar{x}, x \neq \bar{x}} \frac{\langle \gamma, x - \bar{x} \rangle}{\|x - \bar{x}\|} \leq 0 \right\}.$$

- The **Mordukhovich (limiting) normal cone** is defined via the limiting process:

$$N_S(\bar{x}) := \left\{ \gamma \in \mathbb{R}^n \mid \exists x^k \xrightarrow{S} \bar{x}, \gamma^k \to \gamma \text{ s.t. } \gamma^k \in N_S^F(x^k) \text{ for all } k \right\}.$$

– The **Clarke normal cone** is the closed convex hull of the limiting normal cone:

$$K_S(\bar{x}) := \mathrm{cl}(\mathrm{conv}\ N_S(\bar{x})).$$

In particular, when $S = C$ is the convex constraint set appearing in (CCOP), we write $N_C^F(\bar{x})$, $N_C(\bar{x})$, and $K_C(\bar{x})$ for the corresponding Fréchet, Mordukhovich, and Clarke normal cones.

Remark 1 (Normal cones for the sparsity set) For the cardinality set $C_s = \{x \in \mathbb{R}^n : \|x\|_0 \le s\}$, all three normal cones coincide. The equalities

$$N_{C_s}^F(\bar{x}) = N_{C_s}(\bar{x}) = K_{C_s}(\bar{x}) = \{\gamma \in \mathbb{R}^n : \gamma_{I_1(\bar{x})} = 0\}$$

hold for every $\bar{x} \in C_s$. In particular, if $\|\bar{x}\|_0 < s$, then $N_{C_s}^F(\bar{x}) = \{0\}$. This identity, established in [19,22], ensures that the stationarity concepts CC-AM and CC-M for problem (CCOP) reduce to the simple form used throughout our analysis.

Remark 2 (Normal cones for convex sets C) For the convex constraint set C, the situation is simpler: since C is closed and convex, the Fréchet and Mordukhovich normal cones coincide,

$$N_C^F(\bar{x}) = N_C(\bar{x}), \qquad \forall\, \bar{x} \in C.$$

However, equality with the Clarke (cone-continuity) cone, $K_C(\bar{x})$, is *not* automatic. It requires a mild regularity condition such as polyhedrality or the CC-CPLD property [19]. In our analysis, this is the only point where such regularity is invoked: whereas C_s always satisfies $N_{C_s}^F = N_{C_s} = K_{C_s}$, for the convex set C we assume polyhedrality or CC-CPLD whenever we need $N_C^F = N_C = K_C$.

CC-AM Stationarity for (CCOP): A feasible point $\bar{x} \in C \cap C_s$ is CC-AM if there exist sequences $\{x^k\} \subset C$, $\{u^k\} \subset \mathbb{R}^n$, and $\{\gamma^k\} \subset \mathbb{R}^n$ such that

$$x^k \to \bar{x}, \qquad u^k \in N_C^F(x^k)\ \ \forall k, \qquad g(x^k) + u^k + \gamma^k \to 0,$$

and $\gamma^k \in N_{C_s}^F(x^k)$ for all k. Here, N^F denotes the Fréchet normal cone. For convex C, $N_C^F = N_C$, and for the sparsity set C_s, we have $N_{C_s}^F = N_{C_s} = K_{C_s}$.

CC-M Stationarity for (CCOP): A feasible point $\bar{x} \in C \cap C_s$ is called CC-M (M-stationarity) if there exist $u \in N_C(\bar{x})$ and $\gamma \in N_{C_s}(\bar{x})$ such that

$$g(\bar{x}) + u + \gamma = 0.$$

Equivalently,

$$0 \in g(\bar{x}) + N_C(\bar{x}) + N_{C_s}(\bar{x}),$$

where N denotes the Mordukhovich (limiting) normal cone.

Remark 3 (Cone-continuity for symmetric convex sets) Beyond the polyhedral and CC-CPLD cases treated in [19], many constraint sets of practical importance (such as permutation-invariant or signed-symmetric convex bodies) are *type-1* or *type-2* symmetric, i.e., invariant under the action of a finite group of linear isometries. In [19], a cone-continuity-type regularity condition, called CC-AM-regularity, is introduced to relate CC-AM and CC-M stationarity. Abstracting this idea to a purely geometric setting, we say that a closed, convex set C satisfies the **cone-continuity property** (CCP) if the Fréchet normal cone mapping $x \mapsto N_C^F(x)$ is outer semicontinuous on each support face of C. As shown in Lemma 1, every closed, convex symmetric set automatically satisfies CCP. Thus, symmetric convex sets fit naturally into the cone-continuity framework underlying the regularity assumptions of [19], and their normal cone behavior is fully compatible with the stationarity analysis used in this work.

Before proceeding, we establish a structural regularity property of symmetric convex sets that plays a key role in our stationarity analysis. Recall that a cone-continuity-type regularity condition is needed to relate CC-AM and CC-M stationarity, and that this condition is automatically satisfied under CC-CPLD [19, Cor. 4.10(a)]. The next result shows that this cone—CCP also holds under a different and particularly relevant geometric assumption—symmetry of the feasible region. Its proof relies on the equivariance of normal cones under linear isometries and is deferred to [28, Section 4].

Lemma 1 (CCP for Symmetric Convex Sets) *Let $C \subseteq \mathbb{R}^n$ be a closed, convex, type-1 symmetric set, or a type-2 symmetric set. Then the Fréchet normal cone mapping $x \mapsto N_C^F(x)$ is outer semicontinuous on every face of C determined by a fixed support pattern. In particular, C satisfies CCP in the sense of [19].*

CC-AM Regularity and Intersection Calculus. For the sparsity set C_s, the Fréchet normal cone depends only on the support pattern; it is locally constant and satisfies $N_{C_s}^F(\bar{x}) = N_{C_s}(\bar{x}) = K_{C_s}(\bar{x})$. For the convex set C, we always have $N_C^F(\bar{x}) = N_C(\bar{x})$. In the present work, the sets C under consideration

are closed, convex, and symmetric (type-1 or type-2). All symmetric convex sets considered in this work have nonempty interior; hence, a strong regularity condition (e.g., **complementarity-constrained Mangasarian–Fromovitz condition (CC-MFCQ)** in the sense of [19]) holds, which is stronger than CC-CPLD and ensures that CCP holds automatically.

Consequently, for these symmetric sets, we have $N_C^F(\bar{x}) = N_C(\bar{x}) = K_C(\bar{x})$. Moreover, the existence of an interior point for C ensures that the normal cone to the intersection $\Omega = C \cap C_s$ satisfies the basic calculus sum rule:

$$N_\Omega(\bar{x}) \subseteq N_C(\bar{x}) + N_{C_s}(\bar{x}).$$

This provides the mathematical rigour for the stationarity concepts utilized in our analysis, as the Fréchet, Mordukhovich, and Clarke normal cones behave consistently under these conditions.

CC-S Stationarity for (CCOP): A feasible point $\bar{x} \in C \cap C_s$ is called CC-S (strongly stationarity) if there exist $u \in N_C(\bar{x})$ and $\gamma \in N_{C_s}(\bar{x})$ such that:

(i) Exact first-order condition: $g(\bar{x}) + u + \gamma = 0$;
(ii) Activity rule: If $\|\bar{x}\|_0 < s$, then $N_{C_s}(\bar{x}) = \{0\}$ and hence $\gamma = 0$. Otherwise, $\|\bar{x}\|_0 = s$, then $\gamma_{I_1(\bar{x})} = 0$ and $\gamma_{I_0(\bar{x})}$ is unrestricted; i.e., $N_{C_s}(\bar{x}) = \{\gamma \in \mathbb{R}^n : \gamma_{I_1(\bar{x})} = 0\}$.

Equivalently, $0 \in g(\bar{x}) + N_C(\bar{x}) + N_{C_s}(\bar{x})$ with $N_{C_s}(\bar{x}) = \{0\}$ if $\|\bar{x}\|_0 < s$ and $N_{C_s}(\bar{x}) = \{\gamma : \gamma_{I_1(\bar{x})} = 0\}$ if $\|\bar{x}\|_0 = s$.

AW-Stationarity. Ribeiro et al. [33] proposed **approximate weak stationarity** (AW-stationarity) for the (x, y)-reformulation of cardinality-constrained problems, where the sparsity constraint is modeled via orthogonality conditions $x \circ y = 0$ and simple bounds on y. AW-stationarity is a sequential (AKKT-type) necessary condition that holds for all local minimizers without any constraint qualification. In our sparse symmetric setting, the x-space implications of AW align with CC-AM; under CCP, this further implies CC-M. Thus, unlike BF or Lu–Zhang stationarity, AW and CC-AM stationarities provide necessary conditions for local optimality without requiring constraint qualifications.

We next show that every local minimizer of (CCOP) is CC-AM-stationarity. Our proof (see [28, Section 5]) follows the same structure as the argument in [19, Theorem 3.2], but with an important distinction: although any closed, convex set C can in principle be written via (possibly many) inequality constraints, our analysis does not rely on such an explicit representation. Instead, we exploit the geometric structure of C directly. In particular, since we work over a closed, convex, symmetric feasible set rather than the full space $\mathbb{R}^n$, the verification

of the CC-AM conditions becomes simpler than in the general constraint-system setting of [19].

Moreover, CCP is guaranteed under the CC-CPLD condition—the complementarity-constrained extension of the classical CPLD—as established in [19, Cor. 4.10(a)]. Whenever CC-CPLD holds, the implication

$$\texttt{CC-AM} \Rightarrow \texttt{CC-M}$$

follows immediately. In addition, Lemma 1 shows that CCP also holds for every closed, convex symmetric set (type-1 or type-2), thereby enlarging the class of feasible regions for which CC-AM-type cone-continuity regularity—and hence the above implication—is automatically satisfied. The connection between BF points and CC-stationarity is detailed in [28, Section 6].

Theorem 1 (Local minimizers are CC-AM (hence CC-M)) *Let $C \subseteq \mathbb{R}^n$ be closed, convex, and symmetric (either nonnegative type-1 or type-2), and let $\Omega := C \cap C_{\mathrm{s}}$. If $\widehat{x} \in \Omega$ is a local minimizer of f over Ω, then $\widehat{x}$ is* CC-AM *for* (CCOP). *If* CCP *(*AM*-regularity) holds at $\widehat{x}$ (e.g., for polyhedral C or under* CC-CPLD*), then $\widehat{x}$ is in particular* CC-M.

3 An Improved Quasi-Newton Penalty Decomposition Method

In this section, we discuss the algorithmic features of an improved penalty decomposition method, with a focus on its quasi-Newton aspects. In other words, we show how the approximate Hessian of the cost function can be utilized in the penalty decomposition algorithm, resulting in higher accuracy due to the use of second-order model information.

3.1 Main Algorithmic Aspects of the New Approach

As already mentioned, the model (CCOP) is generally NP-hard due to the existence of the cardinality constraint, even when the cost function is quadratic. Consequently, our strategy focuses on handling this intractable constraint as effectively as possible. Notably, our analysis shows that we do not need to decouple the feasibility set from the cardinality constraint when the feasible region is a *symmetric set*. More specifically, for any given point, the problem of finding the closest s-sparse point lying in a symmetric set admits an explicit or low-cost projection rule.

Recall from (1) that

$$C \cap C_{\mathrm{s}} = \{y \in \mathbb{R}^n \mid y \in C,\ \|y\|_0 \leq s\}.$$

Our algorithm is based on the following reformulation of (CCOP):

$$\min_{x\in\mathbb{R}^n,\ y\in C\cap C_{\mathrm{s}}} f(x) \quad \text{s.t.} \quad x-y=0,$$

which can be tackled via a sequence of penalty subproblems as follows:

$$\min_{x\in\mathbb{R}^n,\ y\in C\cap C_{\mathrm{s}}} q_\rho(x,y) := f(x)+\rho\|x-y\|^2.$$

However, due to the (possible) nonconvexity of f, the above penalty subproblem is still expensive to handle. This fact motivates us to suggest the following approximate penalty subproblem:

$$\min_{x\in\mathbb{R}^n,\ y\in C\cap C_{\mathrm{s}}} \Phi_{(\rho,z)}(x,y), \qquad (P_{(x,y;\rho)})$$

where the model function is

$$\Phi_{(\rho,z)}(x,y) := (x-z)^T g(z)+\frac{1}{2}(x-z)^T\mathbf{H}(x-z)+\frac{1}{2}\rho\|x-y\|^2, \qquad (8)$$

in which $\mathbf{H}$ is an approximation of the Hessian $\nabla^2 f(z)$. Since

$$f(x)\approx f(z)+(x-z)^T g(z)+\frac{1}{2}(x-z)^T\mathbf{H}(x-z),$$

it follows that

$$f(z)+\Phi_{(\rho,z)}(x,y)\approx f(x)+\frac{1}{2}\rho\|x-y\|^2.$$

In our analysis, $\mathbf{H}$ is taken as a diagonal approximation of the Hessian of the nonconvex function f at z. This sparse approximation is a notable advantage of our approach, since computing the full (often dense) Hessian is prohibitively expensive.

Although ($P_{(x,y;\rho)}$) is still NP-hard due to the nonconvex cardinality constraint in y, its internal structure lends itself naturally to a block-coordinate treatment, which separates the model into an x-update and a y-update. This decomposition reveals two subproblems with particularly convenient forms: the x-subproblem admits a closed-form minimizer due to the quadratic structure of the model, while the y-subproblem simplifies to a sparse projection onto $C\cap C_s$. These two building blocks form the core of our method and are developed in the following subsections.

3.1.1 Closed-Form Solution of $(P_{(x,y;\rho)})$ with Respect to x

Let j denote the iteration counter of the outer loop, and let ℓ denote the iteration counter of the inner loop. At the jth iteration of PD-QN, the x-subproblem can be written as

$$\min_{x\in\mathbb{R}^n} \ \Phi_{\left(\rho^{(j-1)},x^{(j-1)}\right)}\left(x, y_{\ell-1}^{(j-1)}\right), \tag{P_x}$$

where the model objective function $\Phi_{\left(\rho^{(j-1)},x^{(j-1)}\right)}\left(x, y_{\ell-1}^{(j-1)}\right)$ can be computed by setting $\rho = \rho^{(j-1)}$, $z = x^{(j-1)}$, and $y = y_{\ell-1}^{(j-1)}$ in (8).

Since $\rho^{(j-1)} \geq \rho_{\min}$ and the diagonal Hessian approximation $\mathbf{H}^{(j-1)}$ is safeguarded to remain positive definite, we have $\mathbf{H}^{(j-1)} + \rho^{(j-1)} I \succ 0$, ensuring a unique minimizer. Here, $\rho_{\min}$ is a tuning parameter as a lower bound for ρ^{j-1}. The first-order optimality condition for (P_x) is

$$g\left(x^{(j-1)}\right) - \mathbf{H}^{(j-1)}x^{(j-1)} + \mathbf{H}^{(j-1)}x + \rho^{(j-1)}\left(x - y_{\ell-1}^{(j-1)}\right) = 0. \tag{9}$$

Thus, the closed-form solution of (P_x) is

$$x_\ell^{(j-1)} = \left(\mathbf{H}^{(j-1)} + \rho^{(j-1)} I\right)^{-1}\left(\mathbf{H}^{(j-1)}x^{(j-1)} + \rho^{(j-1)} y_{\ell-1}^{(j-1)} - g\left(x^{(j-1)}\right)\right). \tag{10}$$

3.1.2 Solution of $(P_{(x,y;\rho)})$ with Respect to y

With $x = x_\ell^{(j-1)}$ fixed, the y-subproblem reduces to

$$\min_{y\in C\cap C_s,\ I_1(y)\subseteq\mathcal{L}_\ell} \ \|x_\ell^{(j-1)} - y\|^2, \tag{P_y}$$

where $\mathcal{L}_\ell$ is the candidate support (chosen in line 10 of Algorithm 1, below). This amounts to projecting $x_\ell^{(j-1)}$ onto the restricted feasible set $C \cap C_s$ with support contained in $\mathcal{L}_\ell$.

Depending on the structure of C, this projection admits either an explicit formula (e.g., for $\mathbb{R}^n$, $\mathbb{R}^n_+$, the simplex, or ℓ_p-balls with $p \in \{1, 2, \infty\}$) or can be computed by a simple dedicated routine (e.g., a one-dimensional root search for general ℓ_p with $p \geq 1$, or under box constraints). For completeness, Section 8 in [28, Algorithms 3-6] provides the corresponding procedures. In all cases, the cost is low, and the update is efficient because the projection is restricted to at most s coordinates.

3.2 New Algorithm

We here describe the main structural elements of our proposed quasi-Newton penalty decomposition algorithm (Algorithm 1, below), called `PD-QN`. The method follows the classical penalty decomposition framework, but incorporates several key enhancements that improve its practical performance and theoretical properties. In particular, the inner loop is strengthened by a support-selection mechanism inspired by [4, Algorithm 5], and by two safeguards that **control model descent** and **primal-dual agreement**. Together, these components drive the algorithm toward `BF` points of f while maintaining stability under the symmetry of C. We begin by outlining the role and effect of each modification:

(i) **Support selection in the inner loop.** Lines 9 and 10 of the inner loop of `PD-QN` restrict the second subproblem by imposing $I_1(y) \subseteq \mathcal{L}_\ell$, where $\mathcal{L}_\ell$ is a super-support obtained from the current sparse iterate $y_{\ell-1}^{(j-1)}$ through sorting the vector $-p(-\nabla_x \Phi)$. If the existing support has size s, the method may still adjust the support: indices in the current support can be replaced by new ones corresponding to larger components of the gradient map. If the support is smaller than s, the method enlarges it by selecting indices from $I_0(y_{\ell-1}^{(j-1)})$ with the largest entries of

$$p\Big(-\nabla_x \Phi_{(\rho^{(j-1)},x^{(j-1)})}\Big(x_\ell^{(j-1)}, y_{\ell-1}^{(j-1)}\Big)\Big).$$

This procedure continues until full support is obtained or no further decrease in the model function is detected.

(ii) **Diagonal Hessian approximation.** The Hessian approximation $\mathbf{H}$ is taken diagonal, which keeps the closed-form update (10) computationally low-cost (see [28, Section 7] for how $\mathbf{H}$ can be computed). This choice reduces both storage and inversion costs and is essential for scalability: a dense Hessian would incur $O(n^3)$ operations per update, which is prohibitive for large-scale problems.

(iii) **Hardness of the joint subproblem.** Although $\Phi_{(\rho,z)}$ is quadratic in both variables, the constraint $y \in C \cap C_s$ renders problem $(P_{(x,y;\rho)})$ NP-hard. The block-coordinate decomposition used in Algorithm 1 circumvents this difficulty by splitting variables and exploiting the fact that both subproblems (P_x) and (P_y) admit closed-form or inexpensive solutions.

(iv) **Structure of the y-update.** The projection defining the y-update involves only the s indices in the selected support. Hence, although the ambient dimension may be very large, the projection step effectively operates in a space of dimension s. This property is a major contributor to the scalability of `PD-QN`.

(v) **Υ-based restart safeguard.** The condition

$$\min_x \Phi_{(\rho^{(j)},x^{(j)})}(x, y^{(j)}) \le \Upsilon^{(j-1)}$$

enforces a non-increasing control sequence $\{\Upsilon^{(j)}\}$ and prevents undesirable model increases. If violated, the algorithm restarts with $y_0^{(j)} := y_0^{(0)}$. This restart does not attempt to randomize the initial point; instead, it restores the descent guarantee of the model and ensures stability of the outer iteration.
(vi) **Effect of the Υ-restart.** The restart does not modify the subproblems themselves, but only resets the initialization of y at the next outer step. Since `PD-QN` is deterministic, repeated restarts from the same initial $y_0^{(0)}$ simply enforce the descent safeguard and are consistent with the convergence analysis.
(vii) **Primal-dual agreement safeguard.** The mechanism

$$\|x^{(j)} - y^{(j)}\| \leq \tau \|x^{(j-1)} - y^{(j-1)}\| + \eta_j, \qquad \tau \in (0,1),\ \eta_j \downarrow 0, \tag{11}$$

enforces a contraction of the primal-dual gap. If violated, a restart is triggered (again, resetting $y_0^{(j)} := y_0^{(0)}$), and this condition is instrumental in establishing $\|x^{(j)} - y^{(j)}\| \to 0$ in Theorem 4, below.
(viii) **Effect of the agreement safeguard.** This safeguard complements the approximate stationarity condition and ensures that the primal variable $x^{(j)}$ remains consistent with the sparse projection $y^{(j)}$. It supports the proof of `CC-AM` for limit points, and, under `AM`-regularity assumptions, `CC-M` optimality.

In finite precision arithmetic, at iteration j of `PD-QN`, we regard $x^{(j)} := x_\ell^{(j-1)}$ as an approximate stationarity point of the x-subproblem of the penalty model if

$$\left\| \nabla_x \Phi_{(\rho^{(j-1)}, x^{(j-1)})} \left(x^{(j)}, y^{(j)} \right) \right\| \leq \varepsilon_{j-1}. \tag{12}$$

The outer iterate $(x^{(j)}, y^{(j)})$ is defined as the last inner-loop pair $(x_\ell^{(j-1)}, y_\ell^{(j-1)})$ computed for the model $\Phi_{(\rho^{(j-1)}, x^{(j-1)})}$. In what follows, we describe how the subproblems in x and y (lines 8 and 11 of `PD-QN`) are solved. Both updates admit explicit or near-closed-form solutions: x has a closed-form minimizer, and y is obtained through a sparse projection, with explicit formulas in common cases (full space, orthant, simplex, ℓ_1, ℓ_2, and ℓ_∞) and simple 1-D routines in the remaining ones.

Remark 4 (Support adjustment and restart mechanism) Two aspects of Algorithm 1 deserve clarification. First, regarding the *support-selection rule* in the inner loop, the support is not frozen once it reaches cardinality s. Even when $|I_1(y_{\ell-1}^{(j-1)})| = s$, the algorithm may replace indices in the current support by new ones corresponding to larger components of the gradient map $p(-\nabla_x \Phi)$. Thus, the support set $\mathcal{L}_\ell$ continues to evolve so as to track the most significant coordinates. The case $|I_1(y_{\ell-1}^{(j-1)})| < s$ mainly arises during initialization and is included for completeness. Second, the *restart mechanism* is deterministic and should not be interpreted as a randomization strategy. When either the Υ-based descent safeguard or the primal-dual agreement condition is violated,

the algorithm resets the initialization of the next outer iteration by setting $y_0^{(j)} := y_0^{(0)}$ and restarting the inner loop with $\ell = 0$. Although the restart uses the same initial point, the penalty parameter $\rho^{(j)}$ and the Hessian approximation $\mathbf{H}^{(j)}$ have changed, and hence the underlying model function $\Phi_{(\rho^{(j)}, x^{(j)})}$ is different. The restart therefore acts as a descent safeguard for the evolving penalty model and is essential for the convergence analysis in Theorem 4, below.

Algorithm 1 A Quasi-Newton Penalty Decomposition Algorithm (`PD-QN`) for Solving (`CCOP`)

tuning parameters: $r > 1$, $\widehat{c} > 0$, $\rho_{\max} > \rho^{(0)} > \rho_{\min} > 0$, and sequences $\{\varepsilon_j\}_{j\in\mathbb{N}}$ with $\varepsilon_j \downarrow 0$, and $\{\eta_j\}_{j\in\mathbb{N}}$ with $\eta_j \downarrow 0$, plus an agreement factor $\tau \in (0,1)$.

input: A positive definite matrix $\mathbf{H}^0 \in \mathbb{R}^{n\times n}$, initial points $x_0^{(0)} \in \mathbb{R}^n$, $y_0^{(0)} \in C \cap C_{\mathrm{s}}$.

Compute $f\Big(x_0^{(0)}\Big)$ and find $\Upsilon^{(0)}$ satisfying

$$\Upsilon^{(0)} \geq \max\left\{ f\Big(x_0^{(0)}\Big), \min_{x\in\mathbb{R}^n} \Phi_{(\rho^{(0)}, x_0^{(0)})}\Big(x, y_0^{(0)}\Big), \widehat{c} \right\} > 0. \tag{13}$$

for $j = 1, 2, \ldots$ **do**
  Set $\ell = 0$.
  repeat
    Set $\ell \leftarrow \ell + 1$.
    $x_\ell^{(j-1)} = \operatorname*{argmin}_{x\in\mathbb{R}^n} \Phi_{(\rho^{(j-1)}, x^{(j-1)})}\Big(x, y_{\ell-1}^{(j-1)}\Big)$. [by (10)]
    Choose $\pi \in \tilde{\mathfrak{S}}\Big(-p\Big(-\nabla_x \Phi_{(\rho^{(j-1)}, x^{(j-1)})}\Big(x_\ell^{(j-1)}, y_{\ell-1}^{(j-1)}\Big)\Big)\Big)$.
    Set $i \in [n+1]$ such that $|\mathcal{L}_\ell| = \Big|I_1\big(y_{\ell-1}^{(j-1)}\big) \cup S^\pi_{[i,n]}\Big| = s$.
    $y_\ell^{(j-1)} = \operatorname*{argmin}_{y\in C\cap C_{\mathrm{s}},\ I_1(y)\subseteq\mathcal{L}_\ell} \Phi_{(\rho^{(j-1)}, x^{(j-1)})}\Big(x_\ell^{(j-1)}, y\Big)$.
    $\mathtt{ok} = \Big(\Big\|\nabla_x \Phi_{(\rho^{(j-1)}, x^{(j-1)})}\Big(x_\ell^{(j-1)}, y_\ell^{(j-1)}\Big)\Big\| \leq \varepsilon_{j-1}\Big)$.
  until (`ok`)
  Set $\rho^{(j)} = \max\big(\rho_{\min}, \min(r\cdot\rho^{(j-1)}, \rho_{\max})\big)$.
  Set $\big(x^{(j)}, y^{(j)}\big) = \Big(x_\ell^{(j-1)}, y_\ell^{(j-1)}\Big)$.
  Set `restart` $\leftarrow$ `false`.
  if $\min_{x\in\mathbb{R}^n} \Phi_{(\rho^{(j)}, x^{(j)})}\Big(x, y^{(j)}\Big) > \Upsilon^{(j-1)}$ **then**
    `restart` $\leftarrow$ `true`. % Υ-based reset for safeguarding descent is needed
  end if
  if $j = 1$ **then** $\Delta^{(j-1)} = \|x_0^{(0)} - y_0^{(0)}\|$; **else**, $\Delta^{(j-1)} = \|x^{(j-1)} - y^{(j-1)}\|$; **end if**
  if $\|x^{(j)} - y^{(j)}\| > \tau\,\Delta^{(j-1)} + \eta_j$ **then**
    `restart` $\leftarrow$ `true`. % primal-dual agreement safeguard is needed
  end if
  if `restart` **then** $y_0^{(j)} \leftarrow y_0^{(0)}$; **else**, $y_0^{(j)} \leftarrow y^{(j)}$; **end if**
  Update $\Upsilon^{(j)} = \max\Big\{\Upsilon^{(j-1)},\ f(x^{(j)}),\ \min_{x\in\mathbb{R}^n} \Phi_{(\rho^{(j)}, x^{(j)})}\big(x, y_0^{(j)}\big)\Big\}$.
  Update the Hessian approximation $\mathbf{H}^{(j)}$.
end for

output: $y^{(j)}$

Remark 5 (Finiteness of restart mechanisms) Although Algorithm 1 incorporates two distinct restart safeguards—the Υ-based descent control and the primal-dual agreement condition—both restart mechanisms are invoked only finitely many times along the execution of `PD-QN`. Specifically, the Υ-based restart can occur only finitely often due to the uniform boundedness of the penalty model values under the bounded-penalty regime, while the primal-dual agreement restart is finite as a consequence of the geometric contraction enforced by the agreement condition and the boundedness of the iterates. As a result, after finitely many outer iterations, `PD-QN` proceeds without further restarts, and all subsequent iterates are generated without resetting the inner-loop initialization. Formal proofs of these finiteness properties are provided in Section 4.4.

4 Convergence Analysis

In this section, we develop a convergence theory for `PD-QN`. We begin with a discussion of the bounded-penalty assumption and its role in the algorithmic design.

Bounded penalty regime. Algorithm 1 employs a penalty sequence $\{\rho^{(j)}\}_{j\in\mathbb{N}_0}$ that remains **bounded** between two positive constants. This assumption, formalized in Assumption 3 (penalty parameter bounds and balance condition), below, differs from the classical `PD` and augmented-Lagrangian frameworks (e.g., [18,23]), where the penalty parameter is driven to infinity to enforce feasibility asymptotically. In contrast, `PD-QN` operates in a stabilized, bounded-penalty regime. The bounds $0 < \rho_{\min} \leq \rho^{(j)} \leq \rho_{\max} < \infty$ guarantee uniform spectral conditioning of the quasi-Newton subproblems, permits the safe use of limited-memory updates [11,20], and prevents the numerical ill-conditioning typically observed for large penalties. Feasibility and descent are instead enforced directly by the projection steps onto $C \cap C_{\mathrm{s}}$ and by the primal-dual restart safeguards built into the algorithm, rather than by unbounded growth of $\rho^{(j)}$.

Bounded-penalty strategies of this type have also been successfully employed in proximal and alternating minimization schemes for nonconvex problems, such as PALM [9] and recent ADMM-based quasi-Newton methods [1]. In these approaches, keeping the penalty parameter finite yields better conditioning, allows for accurate quasi-Newton modeling, and facilitates practical convergence in large-scale settings. The following assumption formalizes this bounded-penalty condition and provides the basis for the uniform contraction analysis developed below.

Analytic framework. We analyze the convergence of `PD-QN` in a stabilized **bounded-penalty** regime. Under mild assumptions on the objective function

and the quasi-Newton updates, we first show that all inner and outer iterates are well defined and remain uniformly bounded (Lemma 2, Corollary 2, Proposition 1). As a consequence, the sequence of accepted outer iterates admits accumulation points that satisfy primal feasibility and the imposed sparsity constraint (Theorem 3(i)).

To characterize the limiting behavior of these iterates, we introduce the notion of **asymptotic basic feasible** (ABF) sequences, inspired by the sequential stationarity framework of Kanzow et al. [19]. An ABF sequence is one for which the projected-gradient residuals on the relevant (and possibly evolving) support sets vanish asymptotically (Theorem 3(iii)). Such sequences provide a precise asymptotic description of the iterates produced by PD-QN and may be viewed as sequential approximations of BF points of the original sparse optimization problem (Theorem 4(i)).

We then show that every ABF sequence gives rise to CC-AM stationarity, a first-order necessary condition expressed in terms of the Fréchet normal cones of the convex constraint set and the sparsity set (Theorem 4(ii)). In the present sparse symmetric setting, the cone-continuity property holds automatically (Lemma 1), and therefore CC-AM stationarity implies CC-M stationarity in the sense of Mordukhovich. Combined with the primal-dual agreement safeguard built into PD-QN (Lemma 4), this analysis establishes that accumulation points of the algorithm are both BF and CC-M stationarity for the original problem (Theorem 4, Corollary 3).

Finally, under bounded penalty parameters, the penalty model family enjoys **uniform strong convexity**, which yields global quadratic growth and error bound properties independent of the outer iteration index. Leveraging these structural features together with sufficient descent, relative error control, and the primal-dual agreement safeguard, we strengthen the above subsequential guarantees and establish convergence of the **entire** sequence of iterates (Theorem 5). Although this result can be interpreted within the Kurdyka–Łojasiewicz framework (with exponent $\frac{1}{2}$), no explicit KL assumption is required.

4.1 Required Assumptions for Global Convergence of PD-QN

To establish that our algorithm (PD-QN) converges to a BF point and an M-stationarity point, we begin by summarizing some mild assumptions on the gradient $g(x)$ and the Hessian approximations $\{\mathbf{H}^{(j)}\}_{j\in\mathbb{N}_0}$, where $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

Assumption 1 *(To be imposed on the Hessian approximation) For any $j \in \mathbb{N}_0$, all the eigenvalues of $\mathbf{H}^{(j)}$ belong to the interval $[\lambda_{\min}, \lambda_{\max}]$ with $\lambda_{\min} > 0$.*

The assumption that all eigenvalues of $\mathbf{H}^{(j)}$ lie in $[\lambda_{\min}, \lambda_{\max}]$ ensures uniform positive definiteness. It might help to mention explicitly that this condition rules out ill-conditioning in the diagonal Hessian updates, which is necessary for the strong convexity arguments in Lemma 2. This uniform positive definiteness ensures the strong convexity of each subproblem and precludes numerical ill conditioning of the quasi-Newton updates.

Assumption 2 ***(Gradient growth condition)*** *Let $C \subseteq \mathbb{R}^n$ be closed, convex, and symmetric.*

(i) If C is bounded, let $\zeta := \max\{\|x\| : x \in C\} < \infty$.
(ii) If C is unbounded, let $\zeta \geq 1$ be a fixed reference scaling parameter (e.g., $\zeta := \max\{1, \|y^{(0)}\|\}$).

The gradient $g(x) = \nabla f(x)$ exists and is continuous on $\mathbb{R}^n$, and there exist constants $\gamma > 0$ and $c > 0$ such that, for all $x \in \mathbb{R}^n$,

$$\|g(x)\| \leq \begin{cases} \gamma, & \|x\| \leq c\,\zeta, \\ \gamma\,\|x\|, & \|x\| > c\,\zeta. \end{cases}$$

Here, $c > 0$ is fixed so that the **balance condition**

$$\bar{\kappa} < c(1 - \bar{\theta}) \tag{14}$$

in Assumption 3 holds, where

$$\bar{\theta} := \frac{\lambda_{\max} + \gamma}{\lambda_{\min} + \rho_{\min}}, \qquad \bar{\kappa} := \max\left\{1,\ \frac{\rho_{\min} + \gamma}{\lambda_{\min} + \rho_{\min}},\ \frac{\rho_{\min} + \gamma/\zeta}{\lambda_{\min} + \rho_{\min}}\right\} \tag{15}$$

with the tuning parameters $\rho_{\min}$ and $\rho_{\max}$ satisfying

$$0 < \rho_{\min} \ \leq\ \rho^{(j)} \ \leq\ \rho_{\max} < \infty, \quad \text{for all } j. \tag{16}$$

and the constraints $\lambda_{\min}$ and $\lambda_{\max}$ defined in Assumption 1.

Remark 6 (On the case $C = \infty$ and the Entrance Argument) While Lemma 2, below, is stated for bounded C (where $\zeta < \infty$), the result extends naturally to unbounded sets $C = \mathbb{R}^n$. In the unbounded setting, the parameter ζ no longer represents a global radius of the set, but instead functions as a reference scaling factor (e.g., $\zeta := \max\{1, \|y^{(0)}\|\}$). Under the strict contraction condition $\tilde{\theta} < 1$, the recurrence

$$\|x^{(j)}\| \ \leq\ \bar{\theta}\,\|x^{(j-1)}\| + \bar{\kappa}\,\zeta \tag{17}$$

established in Lemma 2 implies that the sequence $\{x^{(j)}\}_{j \in \mathbb{N}_0}$ is globally attracted to a computational invariant ball $\mathbb{B}(0, R_\infty)$ with radius $R_\infty := \bar{\kappa}\zeta/(1 - \bar{\theta})$, i.e., the **autonomous boundedness property**

$$\limsup_{j \to \infty} \|x^{(j)}\| \leq R_\infty \tag{18}$$

holds (see Corollary 1, below). Specifically, if the initial iterate $x^{(0)}$ lies outside this ball, the geometric decay $\|x^{(j)}\| \leq \tilde{\theta}^j \|x^{(0)}\| +$ constant ensures that the sequence enters the compact region $\mathbb{B}(0, c\zeta)$ in finite time. This "argument" ensures that even without a bounded feasibility set, the trajectory remains within a compact region, where the gradient $g(x)$ is effectively locally Lipschitz, thereby satisfying the necessary conditions for stationarity and global convergence analysis.

Remark 6 currently mentions the unbounded case only briefly. For transparency, it may be worth emphasizing that when C is unbounded, boundedness of iterates relies solely on the contraction inequality (defined by (17), below). This remains consistent with the spirit of the convergence analyses in [22,23], where boundedness of iterates is also ensured, although the mechanism is different: classical PD methods rely on diverging penalties, while in our setting boundedness follows from the strict contraction inequality established in Lemma 2.

The gradient growth condition is indeed weaker than Lipschitz continuity, but it still guarantees local Lipschitz continuity on bounded subsets. In particular, since the iterates are bounded (by Lemma 2), the gradient is effectively Lipschitz continuous along the relevant trajectory. This observation clarifies why standard quasi-Newton arguments remain valid under Assumption 2.

It can be observed that the so-called Lipschitz-from-the-origin condition,

$$\|g(x) - g(0)\| \leq \widehat{\gamma}\|x\|, \quad \text{for all } x \in C \subseteq \mathbb{R}^n,$$

for some constant $\widehat{\gamma} > 0$, implies Assumption 2. Hence, Assumption 2 is weaker than the Lipschitz-from-the-origin condition, which itself is weaker than the standard Lipschitz continuity condition commonly used in the literature. Specifically, the standard Lipschitz condition requires the existence of a constant $L > 0$ such that

$$\|g(x) - g(\widetilde{x})\| \leq L\|x - \widetilde{x}\|, \quad \text{for all } x, \widetilde{x} \in C \subseteq \mathbb{R}^n.$$

Therefore, our assumption on g is significantly less restrictive than the conventional assumptions on the gradient imposed on the relevant literature.

Meanwhile, uniform boundedness of the positive definite Hessian updating formulas is another standard assumption in the convergence analysis of quasi-Newton methods; we here adopt a similar requirement as well.

Assumption 2 also implies that the gradient remains bounded over any bounded subset of $\mathbb{R}^n$. In particular, if C is bounded, then $\|x\| \leq \zeta$ implies $\|g(x)\| \leq \gamma$, so the gradient cannot grow unboundedly along the trajectory of the iterates. This property maintains algorithmic stability and substitutes for the stronger global Lipschitz continuity condition commonly used in quasi-Newton analyses.

Assumption 3 ***(Penalty parameter bounds and balance condition)***

The penalty sequence $\{\rho^{(j)}\}_{j\in\mathbb{N}_0}$ *satisfies* (16) *and the parameters* $\bar{\theta}$ *and* $\bar{\kappa}$ *are defined as* (15)*. With this choice, the unified recurrence* (17) *(proved in Lemma 2, below) satisfies* $(\rho_{\min}\zeta+\gamma)/(\lambda_{\min}+\rho_{\min}) \le \bar{\kappa}\,\zeta$ *for all* $\zeta \ge 1$*. We require:*

(i) ***Strict contraction:*** $\rho_{\min} > \lambda_{\max} + \gamma - \lambda_{\min}$*, equivalently* $0 < \bar{\theta} < 1$.
(ii) ***Balance condition for bounded*** C***:*** *There exists* $c > 0$ *such that the balance condition* (14) *holds.*
(iii) ***Balance condition for unbounded*** C***:*** *If* $C = \mathbb{R}^n$*, the balance condition* (14) *ensures autonomous boundedness of the iterates as quantified by* (18)*.*

Intuitively, the balance condition guarantees that the penalization is strong enough to dominate the possible growth of the gradient term, ensuring a strict contraction of the recurrence into the invariant ball. Under these conditions, the effective contraction factor $\tilde{\theta} := \bar{\theta} + \bar{\kappa}/c$ *satisfies* $\tilde{\theta} < 1$*, which guarantees finite-time entrance into the invariant ball* $\{x : \|x\| \le c\zeta\}$.

The balance condition (14) guarantees that the penalization is strong enough to dominate the possible linear growth of the gradient term $\gamma\|x\|$. In the unbounded case ($C = \mathbb{R}^n$), this condition ensures a **global dissipative property**: the "pull" from the quadratic penalty $\frac{\rho}{2}\|x-y\|^2$ toward the constrained origin outweighs the "push" from the gradient growth. Consequently, the algorithm generates the autonomous boundedness property (18), which provides a surrogate for the compactness of C typically required in penalty methods. Unlike classical methods that drive $\rho \to \infty$ to prevent divergence on unbounded domains, `PD-QN` maintains a bounded $\rho \in [\rho_{\min}, \rho_{\max}]$, preserving the numerical conditioning of the quasi-Newton updates while ensuring stability through this spectral balance.

Remark 7 (Bounded penalty regime versus classical `PD` *methods)* Unlike classical penalty decomposition algorithms analyzed in [18,23] whose convergence analysis relies on an ever-increasing penalty parameter $\rho_k \to \infty$ to drive feasibility, Algorithm 1 operates in a **bounded-penalty regime** with (16). This choice is essential to ensure uniform spectral conditioning of the quasi-Newton subproblems and to establish the contraction inequality of Lemma 2. In the present framework, feasibility and descent are not obtained asymptotically by enlarging the penalty, but are instead enforced directly through the projection steps and the restart safeguards built into the algorithm.

4.2 Uniform upper bound on penalty model values

We now establish that the penalty model values remain uniformly bounded across all outer iterations of `PD-QN`. This property is essential for the convergence analysis, since it guarantees that the safeguard resets in Algorithm 1 always relies on a valid finite bound [2].

In line 3 of Algorithm 1, the initialization computes $\Upsilon^{(0)}$ so that condition (13) holds, ensuring that the very first penalty model is bounded. However, as the algorithm progresses, both the penalty parameter $\rho^{(j)}$ and the iterate $x^{(j)}$ evolve, giving rise to new models of the form

$$x \mapsto \Phi_{(\rho^{(j)},x^{(j)})}\big(x, y_0^{(0)}\big).$$

The initial constant $\Upsilon^{(0)}$ need not control these later models, since the quantities

$$\min_x \Phi_{(\rho^{(j)},x^{(j)})}(x, y_0^{(0)})$$

may increase with j. To address this, Algorithm 1 maintains a nondecreasing sequence of bounds by updating at each outer iteration

$$\Upsilon^{(j)} = \max\left\{\Upsilon^{(j-1)},\ f(x^{(j)}),\ \min_{x\in\mathbb{R}^n} \Phi_{(\rho^{(j)},x^{(j)})}(x, y_0^{(j)})\right\}.$$

This update guarantees that $\Upsilon^{(j)}$ dominates the current function value and the relevant model value, and hence that the reset condition in Algorithm 1 always employs a valid finite threshold.

The following results formalize this property: under Assumptions 1–3, the sequence of penalty models admits a uniform upper bound that is independent of the iteration index.

Lemma 2 (Uniform bound under standard conditions) *Suppose*
• *$f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $g(x) = \nabla f(x)$ satisfies Assumption 2, with constants γ, ζ, c therein.*
• *$C \subseteq \mathbb{R}^n$ is closed, convex, symmetric (either nonnegative type-1 or type-2), and bounded with $\zeta := \max\{\|x\| : x \in C\} < \infty$ (if C is unbounded, ζ is the reference scaling parameter defined in Assumption 2).*
• *The Hessian approximations satisfy Assumption 1: there exist $0 < \lambda_{\min} \le \lambda_{\max}$ such that*

$$\lambda_{\min} I \preceq \mathbf{H}^{(j)} \preceq \lambda_{\max} I \quad \text{for all } j \in \mathbb{N}_0.$$

• *Penalty parameters satisfy Assumption 3, with $\bar{\theta}, \bar{\kappa}$ as in (15) and $\tilde{\theta} < 1$.*
Then:
(i) The unified recurrence (17) holds and therefore the sequence $\{x^{(j)}\}_{j\in\mathbb{N}_0} \subset$

$\mathbb{R}^n$ *is uniformly bounded.*
(ii) There exists a constant $\Upsilon^{exact} > 0$ *such that, for every outer iteration* $j \in \mathbb{N}_0$,

$$\min_{x \in \mathbb{R}^n} \Phi_{(\rho^{(j)}, x^{(j)})}(x, y_0^{(j)}) \;\le\; \Upsilon^{exact}.$$

Proof (i) Boundedness of the iterates $\{x^{(j)}\}_{j \in \mathbb{N}_0}$. From the x-update optimality condition (9), we obtain

$$(\mathbf{H}^{(j-1)} + \rho^{(j-1)} I) x^{(j)} = \mathbf{H}^{(j-1)} x^{(j-1)} + \rho^{(j-1)} \widetilde{y}^{(j-1)} - g(x^{(j-1)}),$$

where $\widetilde{y}^{(j-1)}$ denotes the y paired with the last x-update (in the algorithm $\widetilde{y}^{(j-1)} = y_{\ell-1}^{(j-1)}$). Taking norms and using Assumption 1,

$$\|x^{(j)}\| \;\le\; \frac{1}{\lambda_{\min} + \rho_{\min}} \Big(\lambda_{\max} \|x^{(j-1)}\| + \rho^{(j-1)} \|\widetilde{y}^{(j-1)}\| + \|g(x^{(j-1)})\| \Big).$$

Since $\|\widetilde{y}^{(j-1)}\| \le \zeta$ by boundedness of C, it remains to bound the gradient term. By Assumption 2:

- If $\|x^{(j-1)}\| \le c\zeta$, then $\|g(x^{(j-1)})\| \le \gamma$. Hence

$$\|x^{(j)}\| \;\le\; \frac{1}{\lambda_{\min} + \rho_{\min}} \Big(\lambda_{\max} \|x^{(j-1)}\| + \rho_{\min} \zeta + \gamma \Big). \tag{bd1}$$

- If $\|x^{(j-1)}\| > c\zeta$, then $\|g(x^{(j-1)})\| \le \gamma \|x^{(j-1)}\|$. Hence

$$\|x^{(j)}\| \;\le\; \frac{1}{\lambda_{\min} + \rho_{\min}} \Big((\lambda_{\max} + \gamma) \|x^{(j-1)}\| + \rho_{\min} \zeta \Big). \tag{bd2}$$

Both (bd1) and (bd2) can be cast into the recurrence form

$$\|x^{(j)}\| \;\le\; \theta_j \|x^{(j-1)}\| + \kappa_j \zeta,$$

with

$$\theta_j := \frac{\lambda_{\max} + \gamma}{\lambda_{\min} + \rho^{(j-1)}} \le \bar{\theta}, \qquad \kappa_j := \frac{\rho^{(j-1)} + \gamma}{\lambda_{\min} + \rho^{(j-1)}} \le \bar{\kappa}.$$

Here, $\bar{\theta}$ and $\bar{\kappa}$ are from (15). For the additive constant, we handle the two cases uniformly as follows. In (bd2), it equals $\dfrac{\rho_{\min} \zeta}{\lambda_{\min} + \rho_{\min}}$, and hence is bounded by $\dfrac{(\rho_{\min} + \gamma)\zeta}{\lambda_{\min} + \rho_{\min}}$. In (bd1), it equals $\dfrac{\rho_{\min} \zeta + \gamma}{\lambda_{\min} + \rho_{\min}}$, which does not scale with ζ if $\zeta < 1$. Since $\bar{\kappa}$ in (15) dominates both $\dfrac{\rho_{\min} + \gamma}{\lambda_{\min} + \rho_{\min}}$ and $\dfrac{\rho_{\min} + \gamma/\zeta}{\lambda_{\min} + \rho_{\min}}$, we obtain $\kappa_j \le \bar{\kappa}$ in both (bd1) and (bd2), and hence for all j, (17) is satisfied. Since Assumption 3 only requires the existence of a constant upper bound on

the additive coefficient, we may, without loss of generality, enlarge $\bar{\kappa}$ slightly so that it dominates all admissible values of κ_j. Enlarging $\bar{\kappa}$ preserves the balance condition because we can always choose $c > \bar{\kappa}/(1-\bar{\theta})$.

Since $\zeta \geq 1$ by Assumption 2, the term γ/ζ in (15) remains bounded, ensuring $\bar{\kappa}$ is a finite constant and the unified recurrence (17) is well-defined for both bounded and unbounded sets C.

We now show that $\{x^{(j)}\}_{j\in\mathbb{N}_0}$ enters the ball $\{x : \|x\| \leq c\zeta\}$ in finite time and remains there. Thus, the sequence is uniformly bounded. To do that, we distinguish three cases:

Case ia. **Geometric decay above the threshold:** If $\|x^{(j-1)}\| > c\zeta$, then we show that

$$\|x^{(j)}\| \;\leq\; \tilde{\theta}\,\|x^{(j-1)}\|.$$

From $\|x^{(j-1)}\| > c\zeta$ and (17), we obtain

$$\|x^{(j)}\| \;\leq\; \bar{\theta}\,\|x^{(j-1)}\| + \frac{\bar{\kappa}}{c}\,\|x^{(j-1)}\| = \Big(\bar{\theta} + \frac{\bar{\kappa}}{c}\Big)\|x^{(j-1)}\| = \tilde{\theta}\,\|x^{(j-1)}\|.$$

Here, $\tilde{\theta} = \bar{\theta} + \dfrac{\bar{\kappa}}{c} < 1$ by Assumption 3.

Case ib. **Finite-time entrance:** If $\|x^{(0)}\| > c\zeta$, then after at most

$$\bar{j} \;\leq\; \left\lceil \frac{\log\big(\|x^{(0)}\|/(c\zeta)\big)}{\log(1/\tilde{\theta})} \right\rceil$$

iterations, we show that $\|x^{(\bar{j})}\| \leq c\zeta$. By induction from Case ia, as long as $\|x^{(k)}\| > c\zeta$,

$$\|x^{(j)}\| \;\leq\; \tilde{\theta}^{\,j}\,\|x^{(0)}\|.$$

The bound on $\bar{j}$ ensures $\|x^{(\bar{j})}\| \leq c\zeta$.

Case ic. **Invariance of the ball:** If $\|x^{(j-1)}\| \leq c\zeta$ for some $j \geq 1$, then we show that $\|x^{(j)}\| \leq c\zeta$. Since $\|x^{(j-1)}\| \leq c\zeta$ is assumed, from (17),

$$\|x^{(j)}\| \;\leq\; \bar{\theta}(c\zeta) + \bar{\kappa}\,\zeta \;\leq\; c\bar{\theta}\,\zeta + c(1-\bar{\theta})\,\zeta = c\zeta,$$

where the last inequality uses the balance condition from Assumption 3.

The results of Case ia through Case ic establish a "dissipative" property of the algorithm: the sequence $\{x^{(j)}\}_{j\in\mathbb{N}_0}$ is globally attracted to the ball $\mathbb{B}(0, c\zeta)$ and, once inside, remains trapped within it for all subsequent iterations. Consequently, the entire trajectory resides within the compact set $\mathbb{B}(0, \max\{\|x^{(0)}\|, c\zeta\})$. By the Bolzano-Weierstrass theorem, any sequence contained within a compact subset of $\mathbb{R}^n$ must possess at least one accumulation point. This reduces the convergence analysis on the potentially unbounded

domain $C = \mathbb{R}^n$ to the analysis of a sequence on a compact set, ensuring that the set of limit points $\Omega(\{x^{(j)}\})$ is non-empty.

(ii) Uniform bound on the penalty model values. Fix $j \in \mathbb{N}_0$ and consider

$$\Phi_j(x) := \Phi_{(\rho^{(j)}, x^{(j)})}(x, y_0^{(j)}).$$

Since Φ_j is strongly convex, it admits a unique minimizer x_j^*. By optimality,

$$\min_x \Phi_j(x) = \Phi_j(x_j^*) \;\le\; \Phi_j(x^{(j)}).$$

Evaluating at $x^{(j)}$ gives

$$\Phi_j(x^{(j)}) = \tfrac{1}{2}\rho^{(j)}\|x^{(j)} - y_0^{(j)}\|^2.$$

By part (i), $\|x^{(j)}\| \le c\zeta$, and by boundedness of C, $\|y_0^{(j)}\| \le \zeta$. Hence

$$\|x^{(j)} - y_0^{(j)}\| \;\le\; \|x^{(j)}\| + \|y_0^{(j)}\| \;\le\; (c+1)\zeta.$$

Since $\rho^{(j)} \le \rho_{\max}$, it follows that

$$\min_x \Phi_j(x) \;\le\; \tfrac{1}{2}\rho_{\max}(c+1)^2\zeta^2.$$

Setting $\Upsilon^{\text{exact}} := \frac{1}{2}\rho_{\max}(c+1)^2\zeta^2$ provides the desired uniform bound. □

Corollary 1 (Autonomous boundedness) *Under the assumptions of Lemma 2, the iterates satisfy the condition (18), that is*

$$\limsup_{j\to\infty} \|x^{(j)}\| \;\le\; \frac{\bar{\kappa}\,\zeta}{1-\bar{\theta}} =: R_\infty.$$

Proof From Lemma 2(i), the unified recurrence

$$\|x^{(j)}\| \le \bar{\theta}\|x^{(j-1)}\| + \bar{\kappa}\zeta$$

holds for all j, with $0 < \bar{\theta} < 1$. Unrolling this inequality yields

$$\|x^{(j)}\| \le \bar{\theta}^j\|x^{(0)}\| + \bar{\kappa}\zeta\sum_{i=0}^{j-1}\bar{\theta}^i = \bar{\theta}^j\|x^{(0)}\| + \frac{\bar{\kappa}\zeta}{1-\bar{\theta}}\big(1-\bar{\theta}^j\big).$$

Letting $j \to \infty$ gives the stated bound. □

Remark 8 (On the choice and role of Υ) Lemma 2 provides an explicit uniform bound

$$\Upsilon^{\text{exact}} := \tfrac{1}{2}\rho_{\max}(c+1)^2\zeta^2$$

valid under exact minimization of the x-subproblem. In Proposition 1, this constant is slightly enlarged by an additional error term $\frac{1}{2}(\lambda_{\min} + \rho_{\min})^{-1} \sup_{k\in\mathbb{N}_0} \varepsilon_k^2$ to account for the inexact termination of the inner loop. In practice, however, the exact value of Υ^{exact} (or any enlarged version Υ) is not known a priori. Instead, Algorithm 1 initializes with a safe bound $\Upsilon^{(0)}$ in line 3, chosen to dominate the first penalty model, and then maintains a nondecreasing sequence by updating at each outer iteration:

$$\Upsilon^{(j)} := \max\left\{\Upsilon^{(j-1)}, f(x^{(j)}), \min_{x\in\mathbb{R}^n} \Phi_{(\rho^{(j)},x^{(j)})}\big(x, y_0^{(j)}\big)\right\}.$$

This adaptive safeguard guarantees that each $\Upsilon^{(j)}$ upper-bounds the relevant model values, thereby preserving the correctness of the reset condition at every iteration—without requiring knowledge of the exact constant Υ^{exact}.

Conceptually, the monotonicity of $\{\Upsilon^{(j)}\}_{j\in\mathbb{N}_0}$ plays a role similar to potential functions in descent methods and safeguarding techniques in nonconvex block-coordinate or proximal alternating schemes [9]: it ensures boundedness and robustness of the method even under nonconvexity and inexact subproblem solutions.

Proposition 1 (Uniform Υ on penalty model values) *Let $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ be the sequence generated by* `PD-QN`*, and suppose Assumptions 1–3 hold, together with the conditions of Lemma 2. Define*

$$\Upsilon \;:=\; \frac{1}{2}\rho_{\max}\,(c+1)^2\zeta^2 + \frac{1}{2(\lambda_{\min}+\rho_{\min})}\sup_{k\in\mathbb{N}_0}\varepsilon_k^2,$$

where the constants c, ζ, and $\lambda_{\min}$ are from Assumptions 1–3 and $\rho_{\min}$ and $\rho_{\max}$ are the lower and upper bounds on ρ. Then, for every outer iteration $j \in \mathbb{N}_0$,

$$\Phi_{(\rho^{(j)},x^{(j)})}\big(x^{(j)}, y^{(j)}\big) \;\le\; \Upsilon.$$

Proof We proceed by induction on j. By construction of Algorithm 1 (line 1), the tolerance sequence $\{\varepsilon_j\}_{j\in\mathbb{N}_0}$ satisfies $\varepsilon_j \downarrow 0$, and hence $\sup_{k\in\mathbb{N}_0}\varepsilon_k^2 < \infty$. Therefore, the constant Υ defined above is finite and requires no additional assumption. For $j = 0$, the initialization in line 3 of Algorithm 1 ensures

$$\Phi_{(\rho^{(0)},x^{(0)})}(x^{(0)}, y^{(0)}) \;\le\; \Upsilon,$$

since Υ dominates the initial model bound by construction. Assume the statement holds for some $j \geq 0$. At iteration $j+1$, the reset condition in line 16 of Algorithm 1 yields two possibilities.

Case 1 **(reset triggered).** If $\min_{x \in \mathbb{R}^n} \Phi_{(\rho^{(j)}, x^{(j)})}(x, y^{(j)}) > \Upsilon$, the algorithm resets $y_0^{(j)} := y_0^{(0)}$ and restarts the inner loop. By Lemma 2,

$$\min_{x \in \mathbb{R}^n} \Phi_{(\rho^{(j)}, x^{(j)})}(x, y_0^{(j)}) \leq \tfrac{1}{2}\rho_{\max}(c+1)^2\zeta^2 \leq \Upsilon,$$

so the newly accepted pair $(x^{(j)}, y^{(j)})$ satisfies the required bound.

Case 2 **(no reset).** Then

$$\min_{x \in \mathbb{R}^n} \Phi_{(\rho^{(j)}, x^{(j)})}(x, y^{(j)}) \leq \tfrac{1}{2}\rho_{\max}(c+1)^2\zeta^2 \leq \Upsilon.$$

Moreover, the inner loop terminates only when the x-block residual is small:

$$\left\|\nabla_x \Phi_{(\rho^{(j)}, x^{(j)})}(x^{(j)}, y^{(j)})\right\| \leq \varepsilon_{j-1}.$$

Since $\Phi_{(\rho^{(j)}, x^{(j)})}(\cdot, y^{(j)})$ is $(\lambda_{\min} + \rho_{\min})$-strongly convex, we can invoke the standard error bound for strongly convex functions:

$$\Phi_{(\rho^{(j)}, x^{(j)})}(x^{(j)}, y^{(j)}) \leq \min_x \Phi_{(\rho^{(j)}, x^{(j)})}(x, y^{(j)}) + \frac{\|\nabla_x \Phi_{(\rho^{(j)}, x^{(j)})}(x^{(j)}, y^{(j)})\|^2}{2(\lambda_{\min} + \rho_{\min})}.$$

Therefore

$$\Phi_{(\rho^{(j)}, x^{(j)})}(x^{(j)}, y^{(j)}) \leq \tfrac{1}{2}\rho_{\max}(c+1)^2\zeta^2 + \frac{\varepsilon_{j-1}^2}{2(\lambda_{\min} + \rho_{\min})} \leq \Upsilon,$$

where the last inequality uses the definition of Υ.

Thus, in both cases, the bound is valid at iteration $j+1$. By induction, it holds for every $j \in \mathbb{N}_0$. □

Lemma 3 (Lower bound on quadratic model decrease) *Consider the outer loop iterates of* `PD-QN` *generating the sequence* $\{x^{(j)}\}_{j \in \mathbb{N}_0}$*, where each* $x^{(j)}$ *solves* (P_x) *with the* y *used in its last* x*-update (denote it* $\tilde{y}^{(j)}$*, which in the algorithm equals* $y_{\ell-1}^{(j)}$ *while* $y^{(j)} = y_\ell^{(j)}$*). Assume:*

- *The initial point* $x^{(0)} \in \mathbb{R}^n$ *is arbitrary and* $y^{(j)} \in C_s \cap C$ *for every* $j \in \mathbb{N}_0$.
- $C \subseteq \mathbb{R}^n$ *is closed, convex, symmetric (either nonnegative type-1 or type-2);* ζ *is as defined in Assumption 2, ensuring* ζ *is a valid constant even if* C *is unbounded.*

– *Assumptions 1–3 hold, with* $\bar{\theta}, \bar{\kappa}$ *from* (15) *satisfying* $\bar{\theta} < 1$ *and* $\bar{\kappa} < c(1-\bar{\theta})$ *so that Lemma 2 applies.*

Then the sequence

$$\left\{ (x^{(j)} - x^{(j-1)})^T g(x^{(j-1)}) + \tfrac{1}{2}(x^{(j)} - x^{(j-1)})^T \mathbf{H}^{(j-1)}(x^{(j)} - x^{(j-1)}) \right\}_{j\in\mathbb{N}} \tag{19}$$

is bounded below by some constant $\widehat{c} \in \mathbb{R}$.

Proof By the optimality condition (9) of (P_x) with $y = \widetilde{y}^{(j-1)}$, we have

$$(\mathbf{H}^{(j-1)} + \rho^{(j-1)} I)x^{(j)} = \mathbf{H}^{(j-1)} x^{(j-1)} + \rho^{(j-1)} \widetilde{y}^{(j-1)} - g(x^{(j-1)}).$$

From Lemma 2, the sequence of iterates $\{x^{(j)}\}$ is uniformly bounded and enters the ball $\{x : \|x\| \le c\zeta\}$ after finitely many steps, remaining there thereafter. Let $\widehat{j}$ denote the first index with $\|x^{(\widehat{j})}\| \le c\zeta$. Then, by invariance of the ball, for all $j \ge \widehat{j} + 1$, we have $\|x^{(j-1)}\| \le c\zeta$; hence, Assumption 2 ensures $\|g(x^{(j-1)})\| \le \gamma$.

For any $j \ge \widehat{j}$, define $a := x^{(j)} - x^{(j-1)}$ and $b := g(x^{(j-1)})$. Since $\mathbf{H}^{(j-1)} \succeq \lambda_{\min} I$, we have the inequality

$$a^T b + \frac{1}{2} a^T H^{(j-1)} a \;\ge\; a^T b + \frac{\lambda_{\min}}{2}\|a\|^2 \;\ge\; -\frac{\|b\|^2}{2\lambda_{\min}}.$$

The last step follows from completing the square:

$$a^T b + \frac{\lambda_{\min}}{2}\|a\|^2 = \frac{\lambda_{\min}}{2}\left\| a + \frac{1}{\lambda_{\min}} b \right\|^2 - \frac{1}{2\lambda_{\min}}\|b\|^2.$$

Hence

$$(x^{(j)} - x^{(j-1)})^T g(x^{(j-1)}) + \tfrac{1}{2}(x^{(j)} - x^{(j-1)})^T \mathbf{H}^{(j-1)}(x^{(j)} - x^{(j-1)}) \;\ge c_1,$$

where $c_1 := -\dfrac{\gamma^2}{2\lambda_{\min}}$. For the finitely many indices $1 \le j < \widehat{j}$, we define

$$c_2 := \min_{1\le j<\widehat{j}} \left\{ (x^{(j)} - x^{(j-1)})^T g(x^{(j-1)}) + \tfrac{1}{2}(x^{(j)} - x^{(j-1)})^T \mathbf{H}^{(j-1)}(x^{(j)} - x^{(j-1)}) \right\}.$$

By Lemma 2, the iterates $\{x^{(j)}\}$ are uniformly bounded and remain within the invariant ball for all $j \ge \widehat{j}$, ensuring $\|g(x^{(j-1)})\| \le \gamma$. For these indices, completing the square with $\mathbf{H}^{(j-1)} \succeq \lambda_{\min} I$ yields the uniform lower bound $c_1 := -\gamma^2/(2\lambda_{\min})$. For the finitely many indices $1 \le j < \widehat{j}$, the quadratic

expression in (19) remains finite as it is a continuous function of the iterates evaluated over a bounded domain. Consequently, the entire sequence is bounded below by the finite constant $\widehat{c} := \min\{c_1, c_2\}$, where c_2 is the minimum value attained during the initial phase. This ensures the quadratic model decrease remains stable over the compact trajectory of the algorithm. □

Remark 9 (On bounded and unbounded feasible sets) When the feasible set C is unbounded (for example, $C = \mathbb{R}^n$), the parameter ζ introduced in Assumption 2 does not represent a radius of C. Instead, ζ is a fixed reference scaling parameter, for instance, as in Assumption 2(ii). In this setting, boundedness of the iterate sequence $\{x^{(j)}\}$ does not follow from compactness of the feasible set, but is ensured entirely by the strict contraction property encoded in the unified recurrence (17). More precisely, under the condition $\bar{\theta} < 1$ and the balance requirement (14), that is $\bar{\kappa} < c(1 - \bar{\theta})$, from Assumption 3(ii), the recurrence yields the global bound (18). As a consequence, even when C is unbounded, the iterates are globally attracted to a computationally invariant ball whose radius depends only on the algorithmic constants and the initial scaling parameter ζ. This mechanism replaces the compactness assumptions or diverging penalty parameters commonly used in classical penalty decomposition methods, while preserving numerical conditioning through bounded penalty values.

Remark 10 (On bounded penalty parameters) Lemma 3 and Assumption 3 require the penalty parameters to remain bounded below by some constant $\rho_{\min} > 0$. With the capped-and-floored update rule in line 14 of `PD-QN`,

$$\rho^{(j)} = \max\big(\rho_{\min},\ \min(r \cdot \rho^{(j-1)},\ \rho_{\max})\big),$$

this is automatically satisfied for any prescribed $\rho_{\min}$. The constants $\bar{\theta}$ and $\bar{\kappa}$ from (15) therefore depend only on $\rho_{\min}$ and are given by

$$\bar{\theta} = \frac{\lambda_{\max} + \gamma}{\lambda_{\min} + \rho_{\min}}, \qquad \bar{\kappa} = \max\Big\{ 1,\ \frac{\rho_{\min} + \gamma}{\lambda_{\min} + \rho_{\min}},\ \frac{\rho_{\min} + \gamma/\zeta}{\lambda_{\min} + \rho_{\min}} \Big\}.$$

Thus, to guarantee the contraction and balance conditions in Assumption 3, one simply chooses $\rho_{\min}$ large enough so that $\bar{\theta} < 1$, and then picks $c > \bar{\kappa}/(1 - \bar{\theta})$. Once $\rho_{\min}$ is fixed, $\rho_{\max}$ can be chosen freely (e.g., as a numerical safeguard) without affecting $\bar{\theta}$ or $\bar{\kappa}$.

4.3 Global convergence for the inner loop of `PD-QN`

Before analyzing the outer loop, we first study the inner loop, which alternates between block-coordinate updates of x and y to minimize the penalty-augmented model $\Phi_{(\rho,z)}(x, y)$. Each inner step is a block-coordinate descent

(BCD) update. We first show that the resulting iterates remain bounded, and then establish that the sequence converges to block-coordinate minimizers of the subproblem ($P_{(x,y;\rho)}$). This in turn guarantees that approximate solutions are obtained after finitely many steps at each outer iteration.

Corollary 2 (Boundedness of inner iterates) *There exists $R > 0$ such that*

$$\|x_\ell^{(j)}\| \;\le\; R \qquad \text{for all } j, \ell \in \mathbb{N}_0.$$

Proof By Lemma 2, the outer iterates $\{x^{(j)}\}$ enter the ball $\{x : \|x\| \le c\zeta\}$ in finitely many steps and remain there, hence $\{x^{(j)}\}$ is uniformly bounded. By Assumption 2, $\{g(x^{(j)})\}$ is also bounded whenever $\{x^{(j)}\}$ is bounded. For each fixed j, the inner iterate $x_\ell^{(j-1)}$ is the exact minimizer of a strongly convex quadratic subproblem in x, namely

$$x_\ell^{(j-1)} = (\mathbf{H}^{(j-1)} + \rho^{(j-1)} I)^{-1} \left(\mathbf{H}^{(j-1)} x^{(j-1)} + \rho^{(j-1)} y_{\ell-1}^{(j-1)} - g(x^{(j-1)}) \right).$$

Since $\mathbf{H}^{(j-1)}$ and $\rho^{(j-1)}$ are uniformly bounded and uniformly positive definite (Assumptions 1–3), $\{x^{(j-1)}\}$ is bounded (Lemma 2), $\{y_{\ell-1}^{(j-1)}\} \subset C$ is bounded, and $\{g(x^{(j-1)})\}$ is bounded on bounded sets (Assumption 2), there exists $R > 0$ such that $\|x_\ell^{(j-1)}\| \le R$ for all j, ℓ. □

Theorem 2 (Convergence of the inner loop) *Let $\{(x_\ell^{(j)}, y_\ell^{(j)})\}_{\ell \in \mathbb{N}_0}$ be the sequence generated by the inner loop of* `PD-QN` *for solving the subproblem ($P_{(x,y;\rho)}$). Assume C is bounded. Then:*

(i) The sequence of model values $\{\Phi_{(\rho,z)}(x_\ell^{(j)}, y_\ell^{(j)})\}_{\ell \in \mathbb{N}_0}$ is monotonically non-increasing and convergent.
(ii) Every accumulation point (x^*, y^*) of the inner iterates is a block-coordinate minimizer of ($P_{(x,y;\rho)}$).
(iii) For any prescribed tolerance $\varepsilon_{j-1} > 0$, the inner loop terminates in finitely many steps at each outer iteration.

Proof (i) By construction (lines 8 and 11 of Algorithm 1), each inner update minimizes $\Phi_{(\rho,z)}$ over one block with the other fixed. Thus, for all $\ell \ge 1$,

$$\Phi_{(\rho,z)}(x_\ell^{(j)}, y_\ell^{(j)}) \le \Phi_{(\rho,z)}(x_\ell^{(j)}, y), \qquad \forall y \in C_s \cap C \text{ with } I_1(y) \subseteq \mathcal{L}_\ell, \tag{20}$$

$$\Phi_{(\rho,z)}(x_\ell^{(j)}, y_{\ell-1}^{(j)}) \le \Phi_{(\rho,z)}(x, y_{\ell-1}^{(j)}), \qquad \forall x \in \mathbb{R}^n. \tag{21}$$

Combining (20) and (21) gives

$$\Phi_{(\rho,z)}(x_\ell^{(j)}, y_\ell^{(j)}) \le \Phi_{(\rho,z)}(x_\ell^{(j)}, y_{\ell-1}^{(j)}) \le \Phi_{(\rho,z)}(x_{\ell-1}^{(j)}, y_{\ell-1}^{(j)}),$$

so the model values form a nonincreasing sequence. Lemma 3 ensures that the quadratic decrease term is bounded below, hence the sequence $\{\Phi_{(\rho,z)}(x_\ell^{(j)}, y_\ell^{(j)})\}_{\ell\in\mathbb{N}_0}$ is bounded below and thus convergent.

(ii) Let (x^*, y^*) be an accumulation point along a subsequence $\overline{\mathcal{L}}$. By continuity of $\Phi_{(\rho,z)}$ and closedness of $C \cap C_{\mathrm{s}}$,

$$\Phi_{(\rho,z)}(x^*, y^*) = \lim_{\ell\in\overline{\mathcal{L}}} \Phi_{(\rho,z)}(x_\ell^{(j)}, y_\ell^{(j)}).$$

Taking limits in (20)–(21) gives

$$\begin{aligned}
&\Phi_{(\rho,z)}(x^*, y^*) \le \Phi_{(\rho,z)}(x, y^*), \ \forall x \in \mathbb{R}^n, \\
&\Phi_{(\rho,z)}(x^*, y^*) \le \Phi_{(\rho,z)}(x^*, y), \ \forall y \in C_{\mathrm{s}} \cap C,
\end{aligned}$$

so (x^*, y^*) is feasible and a block-coordinate minimizer.

(iii) Since $\Phi_{(\rho,z)}(\cdot, y_\ell^{(j)})$ is $(\lambda_{\min} + \rho^{(j-1)})$-strongly convex (by Assumptions 1 and 3), for every ℓ, we have

$$\Phi_{(\rho,z)}(x_\ell^{(j)}, y_\ell^{(j)}) - \min_x \Phi_{(\rho,z)}(x, y_\ell^{(j)}) \le \frac{\|\nabla_x \Phi_{(\rho,z)}(x_\ell^{(j)}, y_\ell^{(j)})\|^2}{2(\lambda_{\min} + \rho_{\min})}.$$

Thus, once $\|\nabla_x \Phi_{(\rho,z)}(x_\ell^{(j)}, y_\ell^{(j)})\| \le \varepsilon_{j-1}$, the model suboptimality is at most $\varepsilon_{j-1}^2/(2(\lambda_{\min} + \rho_{\min}))$. If the inner loop did not terminate, the stopping condition would be violated infinitely often and this suboptimality would stay bounded away from zero along an infinite subsequence, while the model values are monotone and convergent by (i), which is a contradiction. Hence the inner loop terminates in finitely many steps for every outer iteration. □

Remark 11 (On block-coordinate minimizers) Theorem 2 shows that accumulation points of the inner loop are block-coordinate minimizers of the subproblem. Although not necessarily global minimizers, they provide a sufficient level of stationarity for the outer-loop analysis: the iterates respect the active sparsity pattern and guarantee descent in the model function. Together with the safeguard mechanisms in `PD-QN`, this ensures subsequential convergence to `BF` and `CC-M` stationarity points of the original problem.

4.4 Global convergence of the outer loop under bounded penalties

In this subsection, we suppose that Assumptions 1–3 hold, with $0 < \rho_{\min} \leq \rho^{(j)} \leq \rho_{\max} < \infty$ for all j. Throughout, $\Phi_{(\rho,z)}(x,y)$ denotes the model function (8), and $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ are the accepted outer iterates of `PD-QN`. We write $\mathcal{L}_j := \operatorname{supp}(y^{(j)})$ and denote by $C_{\mathcal{L}}$ the restriction of C to indices in $\mathcal{L}$. All limits are taken along subsequences, without relabeling.

Having established the convergence of the inner loop, we now turn to the global behavior of the outer loop. At each iteration, `PD-QN` updates the penalty parameter and solves a penalized subproblem designed to enforce both sparsity and feasibility. The central question is whether the sequence of outer iterates accumulates at meaningful limit points of the original problem (`CCOP`). The following results provide a partial answer by showing that any accumulation point satisfies primal-dual agreement ($x^* = y^*$) and the desired sparsity structure, thereby setting the stage for our analysis of basic feasibility and stationarity.

Theorem 3 (Subsequential convergence under bounded penalties) *Assume $C \subseteq \mathbb{R}^n$ is closed, convex, and symmetric (nonnegative type-1 or type-2), and that Assumptions 1–3 hold with bounded penalties. Then the outer sequence admits a subsequence $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ converging to (x^*, y^*) such that*

(i) $y^* \in C \cap C_s$ and $y^*_{\mathcal{L}^c} = 0$ for $\mathcal{L} := \operatorname{supp}(y^*)$ with $|\mathcal{L}| \leq s$;
(ii) for some $\rho^* \in [\rho_{\min}, \rho_{\max}]$,

$$y^*_{\mathcal{L}} \in \operatorname*{argmin}_{v_{\mathcal{L}} \in C_{\mathcal{L}}} \Phi_{(\rho^*, x^*)}(x^*, v), \qquad y^*_{\mathcal{L}^c} = 0,$$

so y^* is a limit point of blockwise minimizers of the inner subproblems;
(iii) the sequence of x-gradients of the penalized models satisfies $\nabla_x \Phi_{(\rho^{(j-1)}, x^{(j-1)})}(x^{(j)}, y^{(j)}) \to 0$, hence $\nabla_x \Phi_{(\rho^*, x^*)}(x^*, y^*) = 0$.

Proof **Boundedness and subsequences.** By Corollary 2, all inner iterates $x_\ell^{(j)}$ are uniformly bounded. In particular, the accepted outer iterates $x^{(j)}$ remain in a bounded set. Proposition 1 then ensures that the corresponding penalty values $\Phi_{(\rho^{(j)}, x^{(j)})}(x^{(j)}, y^{(j)})$ are uniformly bounded as well. Since each $y^{(j)} \in C \cap C_s$ and $y^{(j)}$ is the Euclidean projection of the bounded $x^{(j)}$ onto the closed convex set $C \cap S_{\mathcal{L}_j}$, we have

$$\|y^{(j)}\| \;\leq\; \|x^{(j)}\| + \operatorname{dist}\big(0,\; C \cap S_{\mathcal{L}_j}\big) \;\leq\; \|x^{(j)}\| + \operatorname{dist}(0, C),$$

where $\operatorname{dist}(0, C) < \infty$ because C is nonempty, closed, and convex. Hence, $\{y^{(j)}\}_{j\in\mathbb{N}_0}$ is bounded whenever $\{x^{(j)}\}_{j\in\mathbb{N}_0}$ is bounded. Thus, $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ is bounded, and there exists a convergent subsequence (not relabeled) with

$$(x^{(j)}, y^{(j)}) \;\to\; (x^*, y^*).$$

Because $C \cap C_{\mathrm{s}}$ is closed, we conclude that $y^* \in C \cap C_{\mathrm{s}}$. Moreover, since every $y^{(j)}$ has support of size at most s, so does y^*; thus $y^*_{\mathcal{L}^c} = 0$ with $\mathcal{L} := \operatorname{supp}(y^*)$ and $|\mathcal{L}| \le s$. This proves (i).

Limit block-minimality in y. At every accepted outer iterate, the inner loop computes $y^{(j)} \in \operatorname{argmin}\{\Phi_{(\rho^{(j-1)}, x^{(j-1)})}(x^{(j)}, v) : v \in C \cap C_{\mathrm{s}}, \operatorname{supp}(v) \subseteq \mathcal{L}_j\}$. Since there are finitely many supports of size at most s, an infinite subsequence must eventually repeat one support. Thus, without loss of generality, we may assume $\mathcal{L}_j = \mathcal{L}$ for all indices in the subsequence. By continuity of $\Phi_{(\rho,z)}$ in (ρ, z, x, y) and closedness/convexity of $C_{\mathcal{L}}$, passing to the limit yields $y^*_{\mathcal{L}} \in \operatorname*{argmin}_{v_{\mathcal{L}} \in C_{\mathcal{L}}} \Phi_{(\rho^*, x^*)}(x^*, v)$ with $y^*_{\mathcal{L}^c} = 0$ for some $\rho^* \in [\rho_{\min}, \rho_{\max}]$, proving (ii).

Vanishing model gradient. The inner stopping rule $\|\nabla_x \Phi_{(\rho^{(j-1)}, x^{(j-1)})}(x^{(j)}, y^{(j)})\| \le \varepsilon_{j-1}$ with $\varepsilon_{j-1} \downarrow 0$ implies that the model gradients vanish, giving (iii). □

Theorem 3 describes only sequential properties of the penalized subproblems solved in `PD-QN`. It guarantees boundedness, sparsity preservation, and vanishing model gradients, but it does not yet assert optimality for the original problem. In particular, the result applies only to subsequences of the outer iterates and does not ensure that the primal and dual variables coincide in the limit.

To establish convergence of the entire sequence—and thus stationarity for the original sparse optimization problem—we now invoke the **primal-dual agreement safeguard** built into `PD-QN`. This safeguard enforces geometric decay of the primal-dual discrepancy (lines 20–23 of the `PD-QN` algorithm, which has been defined by (11)), that is

$$\|x^{(j)} - y^{(j)}\| \le \tau \|x^{(j-1)} - y^{(j-1)}\| + \eta_j, \qquad \tau \in (0, 1),\ \eta_j \downarrow 0,$$

which guarantees $\|x^{(j)} - y^{(j)}\| \to 0$ and hence $\{x^{(j)}\}_{j\in\mathbb{N}_0}$ and $\{y^{(j)}\}_{j\in\mathbb{N}_0}$ share a common limit. The next theorem builds on this safeguard to establish that every accumulation point of `PD-QN` is both basic feasible and `CC-M` stationarity for the original problem.

Note that Assumption 3 already enforces the uniform bounds $0 < \rho_{\min} \le \rho^{(j)} \le \rho_{\max}$ for all outer iterations. The following lemma shows that, under these bounds, the restart safeguard of `PD-QN` cannot be triggered infinitely often.

Lemma 4 (Finite restart occurrence) *Under Assumptions 1–3, the restart safeguard used in* PD-QN *can be triggered only finitely many times. In particular, the sequence of accepted outer iterates* $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ *is well-defined for all* j*, and eventually the algorithm proceeds without further restarts.*

Proof A restart occurs only when the tentative pair $(\tilde{x}^{(j)}, \tilde{y}^{(j)})$ fails the agreement condition (11), that here is

$$\|\tilde{x}^{(j)} - \tilde{y}^{(j)}\| \;\le\; \tau\,\|x^{(j-1)} - y^{(j-1)}\| + \eta_{j-1},$$

in which case the penalty parameter is updated according to

$$\rho^{(j)} = \min\{\, r\,\rho^{(j-1)},\ \rho_{\max}\,\}, \qquad r > 1.$$

Thus each restart either:

(a) strictly increases $\rho^{(j)}$ by a factor $r > 1$, or
(b) leaves $\rho^{(j)}$ unchanged because $\rho^{(j-1)} = \rho_{\max}$.

We show that neither case can occur infinitely often.

Case 1: $\rho^{(j)}$ increases. Assumption 3 enforces the uniform bound $0 < \rho^{(j)} \le \rho_{\max} < \infty$ for all j. Since each restart multiplies $\rho^{(j)}$ by $r > 1$, at most

$$N_1 := \left\lceil \frac{\log(\rho_{\max}/\rho_{\min})}{\log r} \right\rceil$$

such restarts can occur before $\rho^{(j)}$ reaches $\rho_{\max}$. Thus, only finitely many restarts fall into Case 1.

Case 2: $\rho^{(j)} = \rho_{\max}$. Once $\rho^{(j)} = \rho_{\max}$, no further increases are possible, so the overall objective model

$$\Phi_{(\rho^{(j)},\, x^{(j-1)})}(x, y)$$

has a *fixed* curvature term $\rho_{\max}\|x-y\|^2/2$ and a uniformly bounded quadratic term $\frac{1}{2}(x-x^{(j-1)})^\top \mathbf{H}(x-x^{(j-1)})$ by Assumption 1. Therefore, the agreement step

$$y^{(j)} = P_{C\cap C_s}(x^{(j)})$$

is a *uniformly contractive* projection in the sense that

$$\|x - y\| \to 0 \quad \text{forces} \quad \|\tilde{x}^{(j)} - \tilde{y}^{(j)}\| \to 0,$$

because both $x^{(j)}$ and $y^{(j)}$ remain in a bounded set (Lemma 2). Hence, when $\rho^{(j)} = \rho_{\max}$, the right-hand side of the agreement condition

$$\tau\|x^{(j-1)} - y^{(j-1)}\| + \eta_{j-1}$$

tends to zero as $j \to \infty$ by the summability of $\{\eta_j\}_{j\in\mathbb{N}_0}$, while the left-hand side $\|\tilde{x}^{(j)} - \tilde{y}^{(j)}\|$ also tends to zero by bounded curvature of the model and Lipschitz continuity of the projection. Thus, the agreement condition is eventually satisfied automatically, and no restart occurs. Hence, Case 2 cannot produce infinitely many restarts.

Note that the value $\rho_{\min}$ does not define a separate case: whenever $\rho^{(j)} < \rho_{\max}$—including the situation $\rho^{(j)} = \rho_{\min}$—a restart always triggers the multiplicative increase $\rho^{(j)} = r\,\rho^{(j-1)}$, and hence this situation is already covered by Case 1.

Both restart types (Cases 1–2) can occur only finitely many times. Thus, the sequence of accepted iterates $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ is well-defined for all j, and after finitely many iterations the algorithm proceeds without further restarts. □

Remark 12 (Finiteness of the restart mechanisms) Algorithm 1 employs two distinct restart safeguards: (i) a *descent-based* restart triggered when the penalty model value exceeds the threshold Υ, and (ii) a *primal-dual agreement* restart triggered when the discrepancy $\|x^{(j)} - y^{(j)}\|$ violates the contraction condition. The descent-based restart can occur only finitely many times. Indeed, Proposition 1 establishes the existence of a uniform constant $\Upsilon < \infty$ such that all accepted penalty model values $\Phi_{(\rho^{(j)},x^{(j)})}(x^{(j)}, y^{(j)})$ are bounded above by Υ. Consequently, the condition that triggers a descent restart cannot be violated infinitely often. The primal-dual agreement restart is also finite. Under the uniform penalty bounds of Assumption 3, each restart either strictly increases the penalty parameter or occurs at $\rho^{(j)} = \rho_{\max}$, where the agreement condition is eventually satisfied automatically due to boundedness of the iterates and geometric decay of the primal-dual discrepancy. This is formalized in Lemma 4. Hence, both restart mechanisms in Algorithm 1 are transient: after finitely many iterations, the algorithm proceeds without further restarts.

Theorem 4 (`BF` and `CC-M` for the original problem) *Under the hypotheses of Theorem 3, let $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ be the outer iterates generated by* `PD-QN`*. Because* `PD-QN` *guarantees the primal-dual agreement safeguard described above, the discrepancy $\|x^{(j)} - y^{(j)}\|$ converges to zero, so that $\{x^{(j)}\}_{j\in\mathbb{N}_0}$ and $\{y^{(j)}\}_{j\in\mathbb{N}_0}$ share a common limit $x^* = y^*$. Then the limit point satisfies:*

(i) $x^* \in C \cap C_s$ and is a `BF` *point of* (`CCOP`).
(ii) There exist $u^* \in N_C^F(x^*)$ and $\lambda^* \in \mathbb{R}^n$ such that

$$g(x^*) + u^* + \lambda^* = 0, \qquad \lambda_i^* = 0 \text{ for all } i \in I_1(x^*),$$

that is, x^* is `CC-AM` *for the original problem. Since C is symmetric (*`CCP` *holds by Lemma 1), x^* is in particular* `CC-M`*.*

Proof By Theorem 3, the outer iterates admit a convergent subsequence $(x^{(j)}, y^{(j)}) \to (x^*, y^*)$ with $y^* \in C \cap C_s$ and $\mathcal{L} := I_1(y^*)$, $|\mathcal{L}| \le s$. Because PD-QN guarantees the primal-dual agreement safeguard, the discrepancy $\|x^{(j)} - y^{(j)}\|$ decreases geometrically and tends to zero. Hence $x^{(j)} - y^{(j)} \to 0$ for the entire sequence, and consequently $x^* = y^*$ and $I_1(x^*) = \mathcal{L}$.

(i) BF **property.** Recall that at every accepted outer iteration, the y-variable is obtained as the exact Euclidean projection of $x^{(j)}$ onto the convex set $C \cap S_{\mathcal{L}_j}$ through the inner-loop y-subproblem. At each accepted outer iterate, the y-block is solved over the convex set $C \cap S_{\mathcal{L}}$, where $S_{\mathcal{L}} := \{v \in \mathbb{R}^n : v_{\mathcal{L}^c} = 0\}$, that is,

$$y^{(j)} \in \operatorname{argmin}\Big\{ \tfrac{\rho^{(j-1)}}{2} \|x^{(j)} - v\|^2 : \; v \in C \cap S_{\mathcal{L}_j} \Big\}, \qquad \mathcal{L}_j := I_1(y^{(j)}).$$

Since only finitely many supports of size at most s exist, along the subsequence we may assume $\mathcal{L}_j = \mathcal{L}$ for all j. Using the fact that each $y^{(j)}$ is a projection of $x^{(j)}$ onto $C \cap S_{\mathcal{L}_j}$ and that $\mathcal{L}_j$ stabilizes along the subsequence, passing to the limit and using $x^* = y^*$ gives

$$y^*_{\mathcal{L}} \in \operatorname*{argmin}_{v_{\mathcal{L}} \in C_{\mathcal{L}}} \; \|x^*_{\mathcal{L}} - v_{\mathcal{L}}\|^2, \qquad y^*_{\mathcal{L}^c} = 0.$$

Because $x^* = y^*$, this is equivalent to $x^*_{\mathcal{L}} = P_{C_{\mathcal{L}}}(x^*_{\mathcal{L}})$ and $x^*_{\mathcal{L}^c} = 0$, i.e., x^* is BF.

(ii) CC-AM **(and** CC-M**).** We combine the first-order conditions of the two blocks.

(B_1) y**-block optimality on** $C \cap S_{\mathcal{L}}$**.** Since C is closed and convex and $S_{\mathcal{L}}$ is a linear subspace of $\mathbb{R}^n$, their intersection $C \cap S_{\mathcal{L}}$ is closed and convex. Hence, optimality of $y^{(j)}$ yields

$$0 \;\in\; \rho^{(j-1)}\big(y^{(j)} - x^{(j)}\big) + N_C^F\big(y^{(j)}\big) + N_{S_{\mathcal{L}}}^F\big(y^{(j)}\big),$$

that is, there exist $u^{(j)} \in N_C^F(y^{(j)})$ and $\lambda^{(j)} \in N_{S_{\mathcal{L}}}^F(y^{(j)})$ such that

$$\rho^{(j-1)}\big(x^{(j)} - y^{(j)}\big) \;=\; u^{(j)} + \lambda^{(j)}. \tag{22}$$

Here, $N_{S_{\mathcal{L}}}^F(y) = \{\lambda \in \mathbb{R}^n : \lambda_{\mathcal{L}} = 0\}$.

(B_2) x**-block approximate stationarity.** By the inner stopping rule,

$$\nabla_x \Phi_{(\rho^{(j-1)}, x^{(j-1)})}\big(x^{(j)}, y^{(j)}\big) \;=\; r^{(j)}, \qquad r^{(j)} \to 0, \tag{23}$$

that is,

$$g\big(x^{(j-1)}\big) + \mathbf{H}^{(j-1)}\big(x^{(j)} - x^{(j-1)}\big) + \rho^{(j-1)}\big(x^{(j)} - y^{(j)}\big) \;=\; r^{(j)}.$$

(B_3) **Substitution and passage to the limit.** Substituting (22) into (23) gives

$$g\big(x^{(j-1)}\big) + \mathbf{H}^{(j-1)}\big(x^{(j)} - x^{(j-1)}\big) + u^{(j)} + \lambda^{(j)} \;=\; r^{(j)}.$$

Along the convergent subsequence, $(x^{(j)}, x^{(j-1)}) \to (x^*, x^*)$, $r^{(j)} \to 0$, and $\mathbf{H}^{(j-1)}\big(x^{(j)} - x^{(j-1)}\big) \to 0$ by boundedness of $\{\mathbf{H}^{(j)}\}_{j\in\mathbb{N}_0}$. By outer semicontinuity of normal cones for closed convex sets, there exist limits $u^* \in N_C^F(x^*)$ and $\lambda^* \in N_{S_\mathcal{L}}^F(x^*)$ of subsequences of $\{u^{(j)}\}_{j\in\mathbb{N}_0}$ and $\{\lambda^{(j)}\}_{j\in\mathbb{N}_0}$, respectively. Hence

$$g(x^*) \;+\; u^* \;+\; \lambda^* \;=\; 0.$$

Since $x^{(j)} \in S_\mathcal{L}$ for all sufficiently large j and $x^{(j)} \to x^*$, we have $x^* \in S_\mathcal{L}$ and hence $\mathcal{L} = I_1(x^*)$ when $\|x^*\|_0 = s$. If $\|x^*\|_0 = s$, then $N_{C_s}^F(x^*) = N_{S_\mathcal{L}}^F(x^*) = \{\lambda : \lambda_\mathcal{L} = 0\}$, so $\lambda^* \in N_{C_s}^F(x^*)$ and x^* is `CC-AM`. If $\|x^*\|_0 < s$, then $N_{C_s}^F(x^*) = \{0\}$ and we may write the stationarity as $g(x^*) + u^* = 0$ with $\lambda^* = 0$, i.e., the `CC-AM` condition with the cardinality multiplier equal to zero. Since C is symmetric, `CCP` holds by Lemma 1, and `CC-AM` implies `CC-M` at x^*. □

Remark 13 (Role of `BFS`*-type support selection)* The `BFS`-type support selection used in the inner loop of `PD-QN` (Algorithm 1, lines 9–10) serves only to generate candidate supports $\mathcal{L}_\ell$ of cardinality at most s and to improve practical performance. The convergence and stationarity analysis depends solely on properties of the accepted outer iterates—namely, blockwise optimality of the y-subproblem, sparsity preservation, and primal-dual agreement—and is therefore independent of the specific mechanism used to select $\mathcal{L}_\ell$. In particular, the proofs remain valid for any support-selection rule that produces supports of size at most s and terminates the inner loop.

Corollary 3 (Convergence of Algorithm 1) *Let $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ be the sequence generated by Algorithm 1. Under Assumptions 1–3, every convergent subsequence has a common limit $x^* = y^*$ such that:*

(i) x^* is a `BF` *point of* (`CCOP`)*;*
(ii) there exist $u^* \in N_C^F(x^*)$ and $\lambda^* \in N_{C_s}^F(x^*)$ with

$$g(x^*) + u^* + \lambda^* = 0 \quad \text{and} \quad \lambda_i^* = 0 \text{ for all } i \in I_1(x^*),$$

that is, x^* is `CC-AM` *for* (`CCOP`)*. Since C is symmetric (hence* `CCP` *holds by Lemma 1), x^* is* `CC-M`*.*

In particular, every convergent subsequence of Algorithm 1 converges to a `BF` *and* `CC-M` *stationarity point of the original sparse optimization problem. The following remarks explain how the primal-dual safeguard in the algorithm ensures the agreement condition $\|x^{(j)} - y^{(j)}\| \to 0$ and how this safeguard acts as an algorithmic enforcement of the* `CC-AM` *regularity property of [19].*

PD-QN enforces primal–dual agreement automatically through a geometric safeguard on $\|x^{(j)} - y^{(j)}\|$. This built-in mechanism replaces the external CC-AM regularity assumption of [19], ensuring that CC-AM implies CC-M without additional constraint qualifications.

4.5 Full-Sequence Convergence under Uniform Strong Convexity (Bounded Penalties)

We now strengthen the subsequential convergence results established above and prove convergence of the *entire* sequence of outer iterates generated by PD-QN. No additional regularity assumptions are required for this result. Indeed, under bounded penalties, each penalty model $\Phi_{(\rho,z)}$ is a uniformly strongly convex quadratic function. This structural property, together with sufficient descent, relative error control, and the primal-dual agreement safeguard built into PD-QN, is sufficient to establish a finite-length property and convergence of the full sequence.

Although uniform strong convexity immediately implies a KL inequality with exponent $\frac{1}{2}$ and uniform constants, we emphasize that no explicit invocation of the KL framework is required. Instead, the analysis below relies directly on the quadratic growth and error bound properties induced by uniform strong convexity.

Theorem 5 (Global convergence under bounded penalties) *Let $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ be the accepted outer iterates produced by PD-QN. Suppose Assumptions 1–3 hold with $0 < \rho_{\min} \le \rho^{(j)} \le \rho_{\max} < \infty$ for all j. Assume further that:*

(i) ***Boundedness.*** *The trajectory is bounded: $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0} \subset \mathcal{K}$ for some compact set $\mathcal{K}$ (cf. Lemma 2 and Proposition 1).*

(ii) ***Sufficient decrease and relative error.*** *Let*

$$\Psi^{(j)} := \Phi_{(\rho^{(j-1)},x^{(j-1)})}(x^{(j)}, y^{(j)}), \qquad F_j(u) := \Phi_{(\rho^{(j-1)},x^{(j-1)})}(u).$$

There exist constants $\delta > 0$, $\kappa \ge 0$, and a summable sequence $\{\varepsilon_j\}_{j\in\mathbb{N}_0}$ introduced in line 1 of PD-QN such that

$$\Psi^{(j)} - \Psi^{(j+1)} \ge \delta \, \|x^{(j+1)} - x^{(j)}\|^2, \tag{24}$$

$$\|\nabla F_j(x^{(j)}, y^{(j)})\| \le \kappa \, \|x^{(j)} - x^{(j-1)}\| + \varepsilon_{j-1}. \tag{25}$$

(iii) ***Agreement safeguard.*** *The primal-dual discrepancy satisfies $\|x^{(j)} - y^{(j)}\| \to 0$ as $j \to \infty$.*

Then the sequence $\{(x^{(j)}, y^{(j)})\}_{j\in\mathbb{N}_0}$ *converges to a single limit* (x^*, y^*)*. Moreover,* $x^* = y^*$*, and by Theorem 4,* x^* *is a* `BF` *point and a* `CC-M` *stationarity point of the original sparse optimization problem.*

Proof Define $u^{(j)} = (x^{(j)}, y^{(j)})$ and

$$F_j(u) := \Phi_{(\rho^{(j-1)}, x^{(j-1)})}(u), \qquad \Psi^{(j)} := F_j(u^{(j)}).$$

By Proposition 1, the sequence $\{\Psi^{(j)}\}_{j\in\mathbb{N}_0}$ is bounded below. Moreover, by construction of the inner loop, each accepted outer iterate satisfies

$$\Psi^{(j+1)} \leq \Psi^{(j)},$$

so the limit

$$\Psi^* := \lim_{j\to\infty} \Psi^{(j)}$$

exists and is finite.

Step 1: Sufficient decrease. Since $\Phi_{(\rho,z)}(\cdot, y)$ is $(\lambda_{\min} + \rho_{\min})$-strongly convex uniformly in (ρ, z), exact minimization of the x-subproblem yields the descent estimate

$$\Psi^{(j)} - \Psi^{(j+1)} \geq \frac{\lambda_{\min} + \rho_{\min}}{2} \|x^{(j+1)} - x^{(j)}\|^2.$$

Thus, the sufficient decrease condition (24) holds with $\delta := \frac{1}{2}(\lambda_{\min} + \rho_{\min})$.

Step 2: Relative error bound. Since F_j is quadratic in (x, y), its gradient ∇F_j is affine and hence Lipschitz continuous on the compact set $\mathcal{K}$; therefore, there exists $\kappa > 0$ such that

$$\|\nabla F_j(u^{(j)}) - \nabla F_j(u^{(j-1)})\| \leq \kappa \|u^{(j)} - u^{(j-1)}\|.$$

Moreover, by the inexact solution of the inner subproblems enforced in `PD-QN`, the residual at iteration $j-1$ satisfies $\|\nabla F_j(u^{(j-1)})\| \leq \varepsilon_{j-1}$. Combining these estimates yields

$$\|\nabla F_j(u^{(j)})\| \leq \kappa \|x^{(j)} - x^{(j-1)}\| + \varepsilon_{j-1},$$

which is the required relative error condition (25).

Step 3: Finite-length property via uniform strong convexity. Uniform strong convexity implies the quadratic growth (error bound) condition

$$F_j(u) - F_j(u^*) \leq \frac{1}{2(\lambda_{\min} + \rho_{\min})} \|\nabla F_j(u)\|^2$$

for every critical point u^* of F_j. Combining this inequality with the sufficient decrease estimate from Step 1 and the relative error bound from Step 2 yields a standard descent recursion, which implies the finite-length property

$$\sum_{j=0}^{\infty} \|x^{(j+1)} - x^{(j)}\| < \infty.$$

Consequently, the sequence $\{x^{(j)}\}_{j\in\mathbb{N}_0}$ is Cauchy and converges to some limit x^*.

Step 4: Agreement and identification of the limit. By the primal-dual agreement safeguard proposed in `PD-QN`, $\|x^{(j)}-y^{(j)}\| \to 0$, and therefore $y^{(j)} \to y^* = x^*$. Finally, Theorem 4 implies that $x^* = y^*$ is a `BF` point and a `CC-M` stationarity point of the original problem. □

5 Numerical Experiments

This section presents a comprehensive numerical study designed to evaluate the practical performance of the proposed penalty decomposition framework for cardinality-constrained optimization. Our experiments aim to assess both the efficiency and robustness of the algorithm in computing high-quality sparse solutions, with particular emphasis on approximate global minimizers and strong (`CC-S`) stationarity points. To ensure a fair and meaningful comparison, we adopt unified stopping criteria, carefully chosen stationarity measures, and standardized computational budgets across all solvers. The proposed method is tested on a broad and diverse benchmark suite, encompassing synthetic and data-driven problems with varying dimensions, sparsity levels, and constraint symmetries. It is compared against several state-of-the-art algorithms using performance profiles based on multiple computational cost measures.

5.1 Stopping Tests and Computational Measures of Stationarity

The stopping tests used in the numerical experiments are based on a combination of objective-based accuracy measures and computable residuals that quantify violations of strong first-order stationarity conditions. These criteria are designed to enable a fair and support-agnostic comparison of algorithms for the nonconvex problem (`CCOP`).

Objective-based accuracy. For a given solver sol $\in \mathcal{S}$, convergence with respect to objective reduction is monitored through the normalized quotient

$$q_{\text{sol}} := \frac{f_{\text{sol}} - f_{\text{opt}}}{f_0 - f_{\text{opt}}}, \tag{26}$$

where f_{sol} denotes the best function value obtained by solver sol, f_0 is the function value at the common initial point, and f_{opt} is the best-known objective value over all solvers. Since f_{opt} is generally unknown in large-scale nonconvex problems, it is approximated in practice by the smallest objective value achieved across all methods under comparison. This normalization allows objective progress to be compared across problems with different scales and conditioning.

Motivation for stationarity-based measures. Problem (CCOP) is nonconvex due to the cardinality constraint and may admit multiple stationarity points lying on different supports. As a result, comparing algorithms solely on the basis of recovered supports or objective values is inherently ill-posed. In particular, different stationarity notions permit different classes of descent directions, and weaker notions may declare convergence at points that are not stationarity in a stronger sense. Consequently, meaningful numerical comparisons must rely on computable residuals that quantify violations of well-defined stationarity conditions in a manner that is independent of the specific support selected by the algorithm.

Throughout this subsection, we consider feasible points $x \in C \cap C_s$. We denote by $I_1(x)$ and $I_0(x)$ the support and off-support index sets, respectively, and by $g(x) = \nabla f(x)$ the gradient of the objective. All stationarity measures are computed using the gradient together with appropriate projections onto the convex set C, accounting for its symmetry and bound structure.

Strong stationarity residual rg_S. We quantify stationarity using a residual $\text{rg}_S(x)$ that measures violation of the strong stationarity condition associated with the inclusion

$$0 \in g(x) + N_C(x) + N_{C_s}(x),$$

as characterized in Section 2.3.2. The residual $\text{rg}_S(x)$ is constructed from two components: (i) a restricted projected-gradient residual on the active support, and (ii) an activity (or swap) violation on the inactive indices. Its precise form depends on whether the cardinality constraint is active and on the symmetry class of the convex set C.

If $\|x\|_0 < s$, the sparsity constraint is inactive and we define

$$\text{rg}_S(x) := \big\| x_S - P_{C_S}\big(x_S - g_S(x)\big) \big\|_\infty,$$

where S denotes the index set of the s largest components of x (in magnitude or in value, depending on the symmetry of C), and P_{C_S} is the projection onto the corresponding restriction of C.

If $\|x\|_0 = s$, the residual additionally accounts for violations of strong optimality conditions on the inactive set and for admissible support swaps, and is defined as

$$\text{rg}_S(x) := \max\Big\{ \big\| x_S - P_{C_S}\big(x_S - g_S(x)\big) \big\|_\infty,\ \text{viol}_{I_0(x)}(x) \Big\},$$

where $\mathrm{viol}_{I_0(x)}(x)$ measures the largest admissible first-order descent associated with activating or swapping an inactive index. Specifically,

$$\mathrm{viol}_{I_0(x)}(x) := \begin{cases} \max\Big\{0,\ \max\limits_{j\in I_0(x)} g_j(x) - \min\limits_{i\in I_1(x)} g_i(x)\Big\}, & \text{for nonnegative type-1 symmetric sets,} \\ \max\Big\{0,\ \max\limits_{j\in I_0(x)} |g_j(x)| - \min\limits_{i\in I_1(x)} |g_i(x)|\Big\}, & \text{for type-2 symmetric sets.} \end{cases}$$

The residual $\mathrm{rg}_S(x)$ vanishes if and only if x satisfies the `CC-S` (strong) stationarity conditions for (`CCOP`), that is, no admissible activation or support swap yields a first-order descent direction.

Stopping criterion. We denote by `nf` the number of function evaluations and by `ng` the number of gradient evaluations, and define `nf2g` = `nf` + 2 `ng`. A problem is considered **solved** by solver sol if there exists a feasible iterate x_{sol} such that the stopping test

$$\mathcal{R}_{\mathrm{sol}} := \begin{cases} q_{\mathrm{sol}}, & \text{if objective-based accuracy is used,} \\ \mathrm{rg}_S(x_{\mathrm{sol}}), & \text{if strong stationarity is targeted,} \end{cases} \qquad \mathcal{R}_{\mathrm{sol}} \le \epsilon$$

is satisfied, and neither the maximum allowed computational budget `nf2gmax` nor the maximum time limit `secmax` is exceeded. Otherwise, the problem is classified as **unsolved**. Since f_{opt} in q_{sol} is defined relative to the set of solvers under comparison, objective-based performance profiles may change when this set is modified, and their interpretation should be understood accordingly. This dependence should be kept in mind when interpreting the objective-based performance profiles in Figures 1–10.

In all numerical experiments reported in this paper, we use `secmax` = ∞, `nf2gmax` = 20000, and accuracy thresholds $\epsilon \in \{10^{-6}, 10^{-3}\}$. This unified stopping framework ensures that solvers are compared consistently in terms of both objective quality and strong stationarity, independently of the support structure reached during optimization.

5.2 Numerical Evaluation and Stationarity-Based Comparison

We now evaluate the practical performance of the proposed penalty decomposition framework across a broad collection of cardinality-constrained optimization problems. Our goals are twofold: (i) to assess the robustness and efficiency of the quasi-Newton variants of `PD-QN` in computing high-quality sparse solutions over a wide range of synthetic and data-driven models, and (ii) to compare these methods with leading state-of-the-art algorithms for sparse

optimization. The benchmark suite includes multiple problem families with both convex and nonconvex objectives, diverse sparsity regimes, and different symmetry structures of the feasible set.

Evaluation challenges. Due to the nonconvex and combinatorial nature of the feasible region $C \cap C_s$, stationarity points are generally **not unique**. Different algorithms may therefore converge to distinct stationarity points, possibly associated with different supports and objective values. As a consequence, no single reference solution or support pattern can serve as a universally meaningful benchmark. This motivates evaluation protocols that combine support-aware diagnostics with support-independent performance measures.

Support-aware and support-independent comparisons. Whenever an exact or globally optimal solution is available, we report support recovery statistics relative to that solution. For problem classes where exact solvers are computationally infeasible, support comparisons are interpreted cautiously and are used only to illustrate qualitative behavior. Crucially, all quantitative performance comparisons are also conducted in a *support-independent* manner using the intrinsic measures q_{sol} and rg_S. This avoids excluding high-quality stationarity points associated with different supports and ensures that algorithms are compared on equal footing.

Dominance analysis. To further highlight differences between algorithms, we perform a dominance-based comparison across stationarity points. An algorithm is said to dominate another on a given problem instance if it achieves both a lower objective value and a smaller strong stationarity residual rg_S, irrespective of the support structure. This analysis captures cases in which `PD-QN` converges to stationarity points of higher quality that would not be identified as superior under support-restricted comparisons alone.

In summary, our numerical evaluation protocol combines (i) objective-based accuracy assessment via q_{sol}, (ii) stationarity-based assessment via the strong residual rg_S, and (iii) dominance analysis that is independent of support patterns. This multi-layered strategy reflects the intrinsic nonconvexity of cardinality-constrained optimization and enables a rigorous, transparent, and fair comparison of algorithms with different convergence behavior and optimality guarantees.

5.3 Test Problems

To comprehensively assess algorithms for cardinality-constrained optimization problems, we employ a unified benchmark generator. The generator produces a diverse collection of both synthetic and data-driven problem classes covering a range of dimensions, sparsity levels, and constraint structures. Each problem is formulated as (`CCOP`). These problems are summarized in Table 1 and their

dimensional and statistical characteristics are summarized in Table 2 (all real datasets are obtained from the UCI Machine Learning Repository[1]). Such problems are also discussed in the survey paper [36].

We generate 30 independent problem instances. For each instance i, the problem dimension is drawn uniformly at random,

$$n_i \sim \mathcal{U}(10,\, 500), \qquad m_i = \max\{2,\, \lfloor 0.5\, n_i \rfloor\},$$

where m_i denotes the number of samples (i.e., the number of rows of A in data-driven models).

The sparsity level s_i is not drawn uniformly but is assigned according to three prescribed sparsity regimes, following the rules implemented in the benchmark generator. With probability 1/3, the instance is labeled *low-sparsity* and we set $s_i = \lfloor 0.15\, n_i \rfloor$. With probability 1/3, the instance is placed in the *medium-sparsity* regime and we set $s_i = \lfloor 0.25\, n_i \rfloor$. The remaining third of the problems are assigned to a *high-sparsity* category, where s_i is chosen as either $\lfloor 0.50\, n_i \rfloor$ or $\lfloor 0.75\, n_i \rfloor$ with equal probability. Finally, to satisfy the theoretical constraints $2 \le s_i < n_i - 1$, we enforce

$$s_i \leftarrow \min\big(\max\{2,\, s_i\},\, n_i - 1\big).$$

This construction yields a balanced set of low-, medium-, and high-sparsity instances across a wide range of problem dimensions.

Each problem is stored as a MATLAB structure containing the relevant data $(A, b, Q, c$ when applicable) together with function handles for f and g. As in [4, Algorithms 1–4], projections and bounds are applied according to the problem type. Two projection symmetries are implemented:

- **Nonnegative-symmetric** (`nneg_sym`): nonnegativity and simplex-type constraints.
- **Absolute-symmetric** (`abs_sym`): sign-symmetric constraints, typical in unconstrained or box-constrained sparse models.

For each problem, a projection operator $\Pi_s(x)$ enforces the sparsity pattern and structural constraints (box, simplex, or unit sphere). The initial vector x^0 is random Gaussian and projected onto the feasible set, i.e.,

$$x^0 = \Pi_s(z), \quad z \sim \mathcal{N}(0, I_n).$$

[1] https://archive.ics.uci.edu/ml/index.php

Table 1 Summary of the cardinality-constrained benchmark problems. `nneg_sym` stands for **nonnegative-symmetric** and `abs_sym` for **absolute-symmetric**. Objectives and gradients are normalized in the benchmark generator.

Problem Type	Objective	Data Structure	Symmetry
Sparse Quadratic	$\frac{1}{2}x^\top Qx + c^\top x$	Random SPD Q	`abs_sym`
Portfolio Optimization	$\frac{1}{2}x^\top Qx - c^\top x$	Toeplitz $Q_{ij} = 0.9^{\lvert i-j\rvert}$	`nneg_sym`
Sparse Regression (Boston)	$\frac{1}{2}\lVert Ax - b\rVert^2$	Boston Housing dataset	`abs_sym`
Logistic Regression (Iris)	$\frac{1}{m}\sum_i \log(1 + e^{-b_i A_i x})$	Binary labels	`abs_sym`
Sparse PCA (Wine)	$-x^\top \Sigma x$	Wine dataset	`abs_sym`
Disjunctive Quadratic	$\frac{1}{2}\lVert Ax\rVert^2$	Random $A \in \mathbb{R}^{\lceil n/2\rceil \times n}$	`abs_sym`
Phase Retrieval	$\frac{1}{2}\lVert\lvert Ax\rvert - b\rVert^2$	Gaussian A	`abs_sym`
Sparse Control Problem	$\frac{1}{2}\lVert Ax - b\rVert^2$	Linear system (A, b)	`abs_sym`

Table 2 Dimensional and statistical characteristics of the generated benchmark problems.

Parameter	Specification
Number of problems	30
Problem dimension	$n \in [10, 500]$ (uniform)
Sample size	$m = \max\{2, \lfloor 0.5\, n\rfloor\}$
Sparsity level	Three-range rule (each with probability 1/3): Low: $s = \lfloor 0.15\, n\rfloor$ Medium: $s = \lfloor 0.25\, n\rfloor$ High: $s \in \{\lfloor 0.50\, n\rfloor, \lfloor 0.75\, n\rfloor\}$ Final adjustment: $s \leftarrow \min(\max\{2, s\}, n-1)$
Real datasets	Iris, Wine, Boston Housing (UCI repository)

Why these test problems are challenging. The benchmark suite considered in this study is deliberately designed to be challenging for sparse optimization algorithms, both algorithmically and theoretically. First, the presence of an explicit cardinality constraint renders all instances combinatorial and nonconvex, with the number of admissible supports growing exponentially in n. Second, many of the problem classes listed in Table 1 exhibit additional sources of nonconvexity beyond sparsity alone, including indefinite quadratic objectives, bilinear structures, and nonsmooth compositions (e.g., absolute values in phase retrieval).

Third, the inclusion of medium- and high-sparsity regimes, where s may be as large as $0.5n$ or $0.75n$, significantly increases difficulty relative to the classical highly sparse setting. In these regimes, the distinction between active and inactive coordinates becomes less pronounced, leading to many competing supports with comparable objective values and weak first-order signals for support selection. This effect is particularly pronounced in symmetric feasible sets, where permutation or sign invariance induces large families of equivalent or nearly equivalent stationarity points.

Finally, for most instances in the benchmark suite, exact global minimizers are unknown and computationally infeasible to obtain. As a result, algorithmic performance cannot be meaningfully assessed solely by support recovery or objective values. Instead, robust evaluation requires stationarity-based criteria

that are independent of the specific support reached, which motivates the use of strong stationarity (`CC-S`) residuals and dominance-based comparisons in our numerical analysis.

5.4 Approximating the Lipschitz constant L

Although the Lipschitz constant of the gradient can, in principle, be computed exactly for several of the quadratic and regression problems in our test set, doing so requires estimating the dominant eigenvalue of matrices such as Q or $A^\top A$. This becomes increasingly expensive as the problem dimension grows, and repeatedly performing such spectral computations inside an iterative method would add substantial overhead with little practical benefit. Moreover, the global constant is typically a very conservative estimate of the local smoothness that actually governs the algorithm's behavior. For these reasons, all algorithms in our study simply start with $L = 1$ and then update it adaptively using a lightweight backtracking rule. At each iteration we compute

$$h = \min\big(0.9,\, \max(10^{-3},\, \max(\|y\|_\infty, 1)\sqrt{\varepsilon})\big), \qquad L \leftarrow \max\big(L,\, |f - f_{\text{old}}|/h\big),$$

which provides a reliable and inexpensive approximation of the local Lipschitz constant. This approach eliminates the need for costly eigenvalue calculations while ensuring that the stepsizes used by the various local solvers remain stable and well-aligned with the local curvature of the objective function.

5.5 Algorithms Compared

We consider the quasi-Newton family associated with Algorithm 1, written compactly as

$$\texttt{PD-QN} \in \{\texttt{PD-D}, \texttt{PD-LM1}, \texttt{PD-LM2}, \texttt{PD-LM3}\},$$

Here, `LM1`, `LM2`, `LM3`, and `D` denote the three limited-memory Hessian approximations and the diagonal approximation, respectively (Algorithms 1–2 in [28]). These algorithms use the enhanced line search scheme described in [28, Section 7]. The final selection of the best-performing variant within this family (`PD-LM1` with `maxiterBFS` $= 250$, called `PD-LM1-a`), based on an extensive comparative study, is deferred to [28, Section 9]. We present the numerical performance of `PD-LM1-a`, the most robust and efficient variant of Algorithm 1, in finding approximate global minimizers and `CC-S` stationarity points. We compare it against the following state-of-the-art methods:

- `IHT`, the iterative hard-thresholding method proposed in [3].
- `PSS`, the sparse-simplex method from [3].

- `GSS`, the greedy sparse-simplex method from [3].
- `BFS`, the basic feasible search method from [4, Algorithm 5].
- `ZCWS`, the zero-`CW` search method from [4, Algorithm 6].

5.6 Practical Enhancements and Safeguards for Algorithm 1

To enhance robustness and practical performance, we incorporate two algorithmic improvements that are applied uniformly to all versions of the proposed method in our numerical comparisons. The first is a warm-start strategy based on `BFS`, which provides a high-quality initial point by identifying a promising sparse support and refining the corresponding coefficients. The second is a stagnation–recovery mechanism that invokes `PSS` only when the inner iterations fail to make progress, thereby enabling the algorithm to escape undesirable stationarity supports. Together, these enhancements improve both efficiency and reliability without altering the underlying structure of the core algorithm.

Warm-start strategy using `BFS`. To initialize our algorithm, we employ `BFS`, which is specifically designed for optimization over sparse symmetric sets. `BFS` provides a high-quality starting point by efficiently identifying a promising support pattern and generating a feasible sparse vector that satisfies the structural constraints of the problem. Warm-starts of this type are widely used in sparse optimization, since the quality of the initial support often has a strong influence on the convergence behavior of subsequent nonconvex methods. In practice, `BFS` frequently locates a support close to optimal, thereby reducing the computational burden on the main solver. Within `BFS`, we further incorporate a restricted `FISTA` step [5] to refine the coefficients on the selected support. This accelerated gradient refinement yields a fast and stable minimization of the objective over a fixed support, resulting in a numerically strong and computationally efficient initial point for the subsequent penalty decomposition iterations. In our experiments, the `BFS` procedure requires at most 15 iterations, although it often terminates much earlier whenever the objective satisfies the condition

$$|f - f_{\text{old}}| < \varepsilon_{\text{BFS}}\,(1 + |f_{\text{old}}|),$$

with $\varepsilon_{\text{BFS}} = 10^{-20}$. A similar rule applies to `FISTA`, which is capped at a finite number of iterations, but typically stops even sooner due to its own convergence criterion. For the number of iterations in [28], a self-tuning has been done.

Stagnation Recovery via `PSS`. When the condition $\|y_{\text{new}} - y_{\text{old}}\| < 10^{-10}$ occurs, it indicates that neither the `PD-QN` update nor the adaptive support; exploration step can modify the current iterate in a meaningful way (here y_{new} and y_{old} were evaluated by line 11 of Algorithm 1 in the current and old

iterations). In practice, this situation means that the algorithm has become stuck at a locally stationarity but suboptimal support, where the penalized gradient direction is too weak and the quasi–Newton correction cannot generate progress. To escape from such cases, we invoke the `PSS` method as a last–resort perturbation. The `PSS` scheme performs simple support–grow or support–swap moves followed by one-dimensional coordinate refinements, and is therefore effective at nudging the iterate out of a poor support. Importantly, we do not use `PSS` as an initialization tool here; instead, it is applied only when the inner `PD-QN` iterations exhibit complete stagnation. This makes its use lightweight and targeted: a single `PSS` step often provides just enough variation in the support for `PD-QN` to resume descent using curvature information and primal–dual consistency. In our experiments, this selective use of `PSS` significantly improves robustness and helps the method avoid getting trapped in undesirable stationarity supports that smooth updates alone cannot overcome. In our experiments, the `PSS` procedure requires at most 5 iterations, although it often terminates much earlier whenever the objective satisfies the condition

$$|f - f_{\text{old}}| < \varepsilon_{\text{PSS}}\,(1 + |f_{\text{old}}|),$$

with $\varepsilon_{\text{PSS}} = 10^{-3}$. A similar rule applies to `FISTA`, which is capped at 5 iterations to solve one-dimensional problems. In `FISTA`, the backtracking procedure used to update the approximate Lipschitz constant L is also limited to at most 5 inner iterations.

5.7 Values for Tuning Parameters

In addition to the parameters associated with the `FISTA`, `PSS`, and `BFS` subroutines, the remaining tuning constants in Algorithm 1 were set as follows: $r = 1.15$, $c = 10^{-8}$, $\widehat{c} = 100$, $\rho^{(0)} = 10^{-2}$, $\rho_{\min} = 10^{-2}$, $\rho_{\max} = 10^{2}$, $\tau = 0.999$, $\varepsilon_{\min} = \varepsilon_m$, $\varepsilon_0 = 0.1$, $m = 10$, $\mu = 1$, and $\varrho = 10^{-10}$. Three parameters are updated during the iterations: the accuracy tolerances ε_j and η_j are set via

$$\varepsilon_j = \eta_j = \max\{\varepsilon_{\min},\, 0.1\, e^{-10^{-3}k}\}, \qquad \varepsilon_{\min} = \varepsilon_m,$$

while the penalty parameter is increased according to

$$\rho^{(j)} = r\,\rho^{(j-1)} \in [\rho_{\min},\, \rho_{\max}].$$

For all competing algorithms, we used their default values for tuning parameters.

5.8 Performance Profile, Efficiency, and Robustness

To compare the efficiency and robustness of our algorithm with the mentioned state-of-the-art algorithms, the performance profile [14] is used based on the two cost measures `nf2g` and `sec`. A solver is considered **most robust** when it successfully handles the largest number of test problems, and **most efficient** when it requires the fewest number `nf2g` of function and gradient evaluations, or the least computational time `sec`. Using `nf2g` as a cost measure is often appropriate because it balances function calls with the typically higher cost of gradient evaluations; in many real-life applications, the gradient is considerably more expensive to compute, so weighting it more heavily yields a more faithful estimate of the actual computational effort. This is particularly relevant for the diverse benchmark problems summarized in Table 1, where gradient computations vary markedly across quadratic, regression, logistic, and phase retrieval models.

5.9 Comparison with State-of-the-Art Algorithms

In this subsection, we compare the performance of the best version `PD-LM1-a` of our algorithm with several state-of-the-art methods, namely `IHT`, `BFS`, `ZCWS`, `PSS`, and `GSS`. The comparison is carried out using performance profiles based on two computational cost measures: `nf2g` and `sec`. For each competing method, we assess efficiency and robustness in computing both approximate global minimizers and `CC-S` stationarity points, under different accuracy requirements. The following subsubsections report and discuss the corresponding results in detail.

5.9.1 `PD-LM1-a` *versus* `IHT`

From Figures 1-2, `PD-LM1-a` is more efficient with respect to the two cost measures `nf2g` and `sec` and more robust than `IHT` for computing both approximate global minimizers and `CC-S` stationarity points.

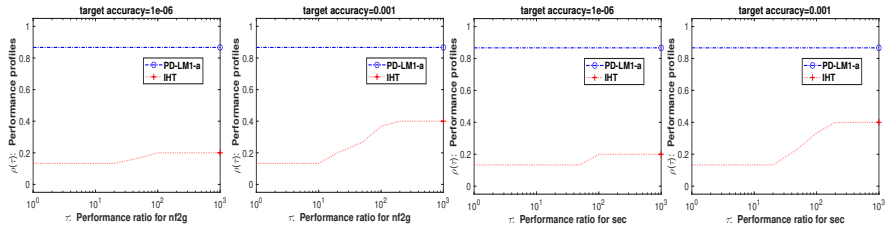

Fig. 1 Performance profiles of `PD-LM1-a` and `IHT` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $q_{\rm sol} \leq 10^{-6}$ (first and third columns) and $q_{\rm sol} \leq 10^{-3}$ (second and fourth columns).

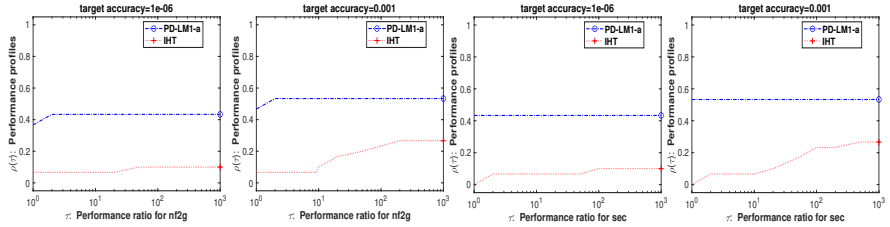

Fig. 2 Performance profiles of `PD-LM1-a` and `IHT` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $\mathrm{rg}_S(x_{\rm sol}) \leq 10^{-6}$ (first and third columns) and $\mathrm{rg}_S(x_{\rm sol}) \leq 10^{-3}$ (second and fourth columns).

5.9.2 `PD-LM1-a` *versus* `BFS`

From Figure 3, `PD-LM1-a` exhibits higher efficiency than `BFS` with respect to both cost measures, `nf2g` and `sec`, and demonstrates greater robustness in computing approximate global minimizers.

From Figure 4, for high accuracy $\epsilon = 10^{-6}$, `PD-LM1-a` outperforms `BFS` in terms of the cost measures `nf2g` and `sec`, and is also more robust for computing `CC-S` stationarity points. For lower accuracy $\epsilon = 10^{-3}$, `PD-LM1-a` remains more efficient than `BFS` with respect to `nf2g` and `sec`, while achieving comparable robustness.

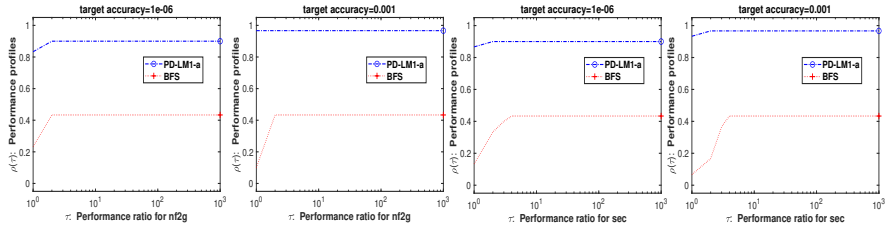

Fig. 3 Performance profiles of `PD-LM1-a` and `BFS` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $q_{\text{sol}} \leq 10^{-6}$ (first and third columns) and $q_{\text{sol}} \leq 10^{-3}$ (second and fourth columns).

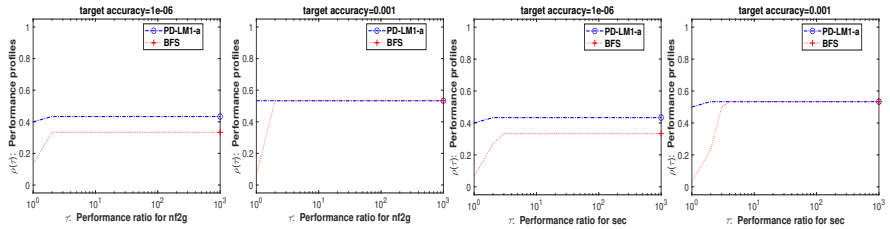

Fig. 4 Performance profiles of `PD-LM1-a` and `BFS` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $\text{rg}_S(x_{\text{sol}}) \leq 10^{-6}$ (first and third columns) and $\text{rg}_S(x_{\text{sol}}) \leq 10^{-3}$ (second and fourth columns).

5.9.3 `PD-LM1-a` *versus* `ZCWS`

From Figure 5, `PD-LM1-a` demonstrates higher efficiency than `ZCWS` with respect to both cost measures, `nf2g` and `sec`, and shows greater robustness in computing approximate global minimizers.

From Figure 6, for high accuracy $\epsilon = 10^{-6}$, `PD-LM1-a` outperforms `ZCWS` in terms of the cost measures `nf2g` and `sec`, and is also more robust in computing `CC-S` stationarity points. For lower accuracy $\epsilon = 10^{-3}$, `PD-LM1-a` remains more efficient than `ZCWS` with respect to `sec`, whereas `ZCWS` is more efficient than `PD-LM1-a` with respect to `nf2g`. Moreover, for both low and high accuracy levels, `PD-LM1-a` exhibits greater robustness than `ZCWS`.

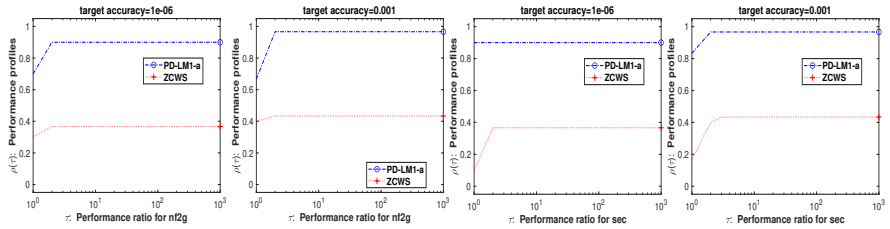

Fig. 5 Performance profiles of `PD-LM1-a` and `ZCWS` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $q_{\rm sol} \leq 10^{-6}$ (first and third columns) and $q_{\rm sol} \leq 10^{-3}$ (second and fourth columns).

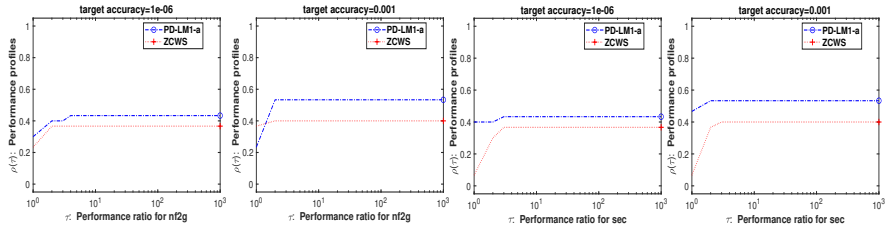

Fig. 6 Performance profiles of `PD-LM1-a` and `ZCWS` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $\mathrm{rg}_S(x_{\rm sol}) \leq 10^{-6}$ (first and third columns) and $\mathrm{rg}_S(x_{\rm sol}) \leq 10^{-3}$ (second and fourth columns).

5.9.4 `PD-LM1-a` *versus* `PSS`

From Figures 7-8, `PD-LM1-a` is more efficient with respect to the two cost measures `nf2g` and `sec` and more robust than `PSS` for computing both approximate global minimizers and `CC-S` stationarity points.

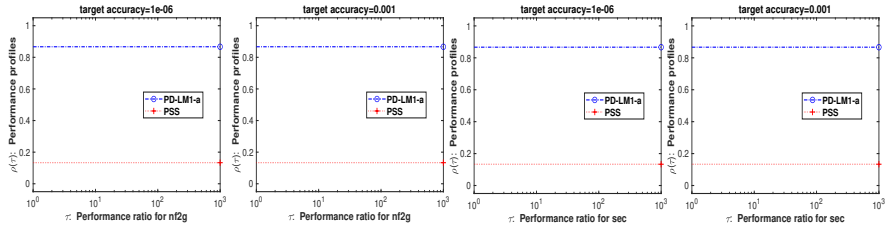

Fig. 7 Performance profiles of `PD-LM1-a` and `PSS` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $q_{\rm sol} \leq 10^{-6}$ (first and third columns) and $q_{\rm sol} \leq 10^{-3}$ (second and fourth columns).

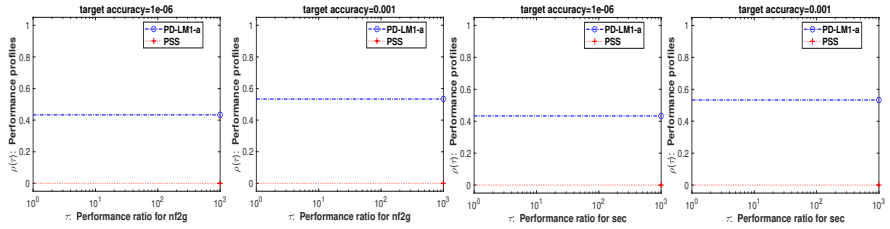

Fig. 8 Performance profiles of `PD-LM1-a` and `PSS` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $\mathrm{rg}_S(x_{\rm sol}) \leq 10^{-6}$ (first and third columns) and $\mathrm{rg}_S(x_{\rm sol}) \leq 10^{-3}$ (second and fourth columns).

5.9.5 `PD-LM1-a` *versus* `GSS`

From Figures 9-10, `PD-LM1-a` is more efficient with respect to the two cost measures `nf2g` and `sec` and more robust than GSS for computing both approximate global minimizers and `CC-S` stationarity points.

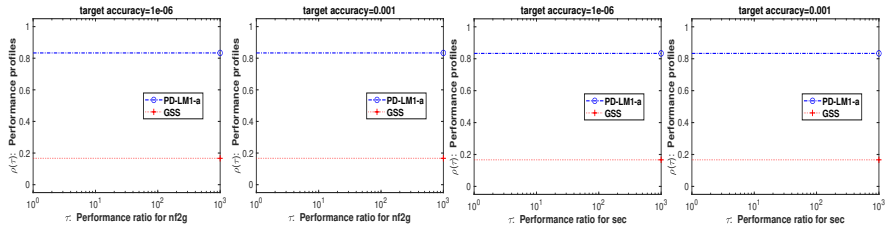

Fig. 9 Performance profiles of `PD-LM1-a` and `GSS` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $q_{\rm sol} \leq 10^{-6}$ (first and third columns) and $q_{\rm sol} \leq 10^{-3}$ (second and fourth columns).

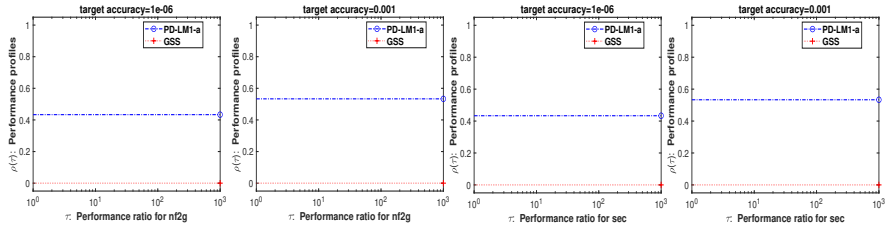

Fig. 10 Performance profiles of `PD-LM1-a` and `GSS` in terms of `nf2g` (first and second columns) and `sec` (third and fourth columns), and with $\mathrm{rg}_S(x_{\rm sol}) \leq 10^{-6}$ (first and third columns) and $\mathrm{rg}_S(x_{\rm sol}) \leq 10^{-3}$ (second and fourth columns).

6 Conclusion

In this paper, we proposed an inexact penalty decomposition algorithm for minimization over sparse symmetric sets. The method is based on a two-block decomposition scheme applied to a sequence of penalized subproblems. At each iteration, the first subproblem is solved in closed form with respect to the primal variable, without imposing sparsity or symmetry constraints, while the second subproblem enforces sparsity and symmetry through an explicit projection onto a restricted feasible set. This structure yields a computationally efficient framework that separates smooth model-based updates from sparse projection steps.

To enable scalability in large-scale settings, we introduced four low-cost diagonal Hessian approximation schemes. Three of these are based on limited-memory information obtained from differences of recent iterates and gradients, while the fourth exploits a controlled distribution of diagonal entries to promote numerical stability. Extensive numerical experiments demonstrate that, with these approximations, the proposed method is competitive with several state-of-the-art algorithms, including `IHT` [3], `PSS` [3], `GSS` [3], `BFS` [4], and `ZCWS` [4].

In finite-precision arithmetic, we employed an enhanced line search strategy based on either backtracking or extrapolation. This procedure evaluates the quadratic model at trial points while computing the true objective only at accepted steps, thereby ensuring sufficient model decrease at low computational cost and improving robustness near stationarity points.

From an algorithmic perspective, we incorporated a `BFS` warm-start strategy, optionally refined by a restricted `FISTA` step, to generate strong initial supports. To mitigate stagnation effects in difficult nonconvex landscapes, a lightweight `PSS` perturbation is invoked selectively to introduce small support modifications, allowing the algorithm to escape unfavorable stationarity regions and resume stable convergence.

From a theoretical standpoint, we established global convergence results under a new gradient growth condition that is strictly weaker than Lipschitz continuity from the origin. Under this assumption and a bounded-penalty regime, every accumulation point of the outer iteration sequence is shown to be both basic feasible and cardinality-constrained Mordukhovich (`CC-M`) stationarity for the original problem. These guarantees bridge the gap between practical penalty decomposition schemes and the strongest available first-order optimality theory for cardinality-constrained optimization.

The numerical results further indicate that, in finite-precision arithmetic and across a wide range of test problems and accuracy requirements, the proposed method consistently produces high-quality sparse solutions. Using objective-based accuracy measures and intrinsic strong stationarity residuals, the algorithm exhibits favorable efficiency and robustness with respect to both computational cost measures considered, namely `nf2g` and `sec`. Overall, the proposed quasi-Newton penalty decomposition framework provides a robust and scalable approach for structured sparse optimization, combining strong theoretical guarantees with competitive practical performance.

Acknowledgements We are grateful to Nadav Hallak for providing the MATLAB implementations of the algorithms described in [4].

Funding The second author acknowledges financial support of the Austrian Science Foundation under https://doi.org/10.55776/PAT2747625.

References

1. Zohre Aminifard and Saman Babaie–Kafaki. A nonmonotone ADMM-based diagonal quasi-Newton update with application to the compressive sensing problem. *Mathematical Modelling and Analysis*, 28(4):673–688, October 2023.
2. Amir Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. SIAM, Philadelphia, 2014.
3. Amir Beck and Yonina C. Eldar. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization*, 23(3):1480–1509, January 2013.
4. Amir Beck and Nadav Hallak. On the minimization over sparse symmetric sets: Projections, optimality conditions, and algorithms. *Mathematics of Operations Research*, 41(1):196–223, February 2016.
5. Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, January 2009.
6. Matteo Bergamaschi, Andrea Cristofari, Vyacheslav Kungurtsev, and Francesco Rinaldi. Probabilistic iterative hard thresholding for sparse learning. *Computational Optimization and Applications*, 93(1):57–83, August 2025.
7. Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2), April 2016.
8. Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, November 2009.
9. Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1–2):459–494, July 2014.
10. Oleg P. Burdakov, Christian Kanzow, and Alexandra Schwartz. Mathematical programs with cardinality constraints: Reformulation by complementarity-type conditions and a regularization method. *SIAM Journal on Optimization*, 26(1):397–425, January 2016.
11. Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, September 1995.
12. Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, January 1998.
13. Aleksandar Cvetković and Vladimir Yu Protasov. The greedy strategy for optimizing the Perron eigenvalue. *Mathematical Programming*, 193(1):1–31, October 2022.
14. Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2, Ser. A):201–213, 2002.
15. Zhengshan Dong and Wenxing Zhu. An improvement of the penalty decomposition method for sparse approximation. *Signal Processing*, 113:52–60, August 2015.
16. Hamid Esmaeili, Shima Shabani, and Morteza Kimiaei. A new generalized shrinkage conjugate gradient method for sparse recovery. *Calcolo*, 56(1), December 2018.
17. Yaohua Hu, Xinlin Hu, and Xiaoqi Yang. On convergence of iterative thresholding algorithms to approximate sparse solution for composite nonconvex optimization. *Mathematical Programming*, 211(1):181–206, March 2025.
18. Christian Kanzow and Matteo Lapucci. Inexact penalty decomposition methods for optimization problems with geometric constraints. *Computational Optimization and Applications*, 85(3):937–971, March 2023.
19. Christian Kanzow, Andreas B. Raharja, and Alexandra Schwartz. Sequential optimality conditions for cardinality-constrained optimization problems with applications. *Computational Optimization and Applications*, 80:185–211, July 2021.
20. Morteza Kimiaei, Arnold Neumaier, and Behzad Azmi. LMBOPT: A limited memory method for bound-constrained optimization. *Mathematical Programming Computation*, 14(2):271–318, January 2022.

21. Matteo Lapucci. A penalty decomposition approach for multi-objective cardinality-constrained optimization problems. *Optimization Methods and Software*, 37(6):2157–2189, April 2022.
22. Matteo Lapucci, Tommaso Levato, and Marco Sciandrone. Convergent inexact penalty decomposition methods for cardinality-constrained problems. *Journal of Optimization Theory and Applications*, 188(2):473–496, December 2020.
23. Zhaosong Lu and Yong Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization*, 23(4):2448–2478, January 2013.
24. Zhaosong Lu, Yong Zhang, and Xiaorui Li. Penalty decomposition methods for rank minimization. *Optimization Methods and Software*, 30(3):531–558, August 2015.
25. Boris S. Mordukhovich. *Variational Analysis and Generalized Differentiation I*. Springer Berlin Heidelberg, 2006.
26. Boris S. Mordukhovich. *Variational Analysis and Applications*. Springer, Cham, 2018.
27. Ahmad Mousavi, Morteza Kimiaei, Saman Babaie–Kafaki, and Vyacheslav Kungurtsev. A class of diagonal quasi-Newton penalty decomposition algorithms for sparse bound-constrained nonconvex optimization. *Optimization Online*, January 2025. Optimization Online Technical Report 29052.
28. Ahmad Mousavi, Morteza Kimiaei, Saman Babaie–Kafaki, and Vyacheslav Kungurtsev. Supplementary material for *An efficient penalty decomposition algorithm for minimization over sparse symmetric sets*. https://github.com/GS1400/suppMat_CC_PDQN, 2025. Supplementary material; accessed: 2025-11-30.
29. Ahmad Mousavi and George Michailidis. Cardinality constrained mean-variance portfolios: A penalty decomposition algorithm. *Computational Optimization and Applications*, 90(3):631–648, January 2025.
30. Arnold Neumaier and Morteza Kimiaei. An improvement of the Goldstein line search. *Optimization Letters*, 18(6):1313–1333, April 2024.
31. Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2006.
32. Andrei Patrascu and Ion Necoara. Penalty decomposition method for solving ℓ_0 regularized problems: Application to trend filtering. In *18th International Conference on System Theory, Control and Computing (ICSTCC)*, page 737–742. IEEE, October 2014.
33. Ademir A. Ribeiro, Mael Sachine, and Evelin H. M. Krulikovski. A comparative study of sequential optimality conditions for mathematical programs with cardinality constraints. *Journal of Optimization Theory and Applications*, 192:1067–1083, February 2022.
34. Kirill Spiridonov, Sergei Sidorov, and Michael Pleshakov. Weak penalty decomposition algorithm for sparse optimization in high dimensional space. In Dmitry Balandin, Konstantin Barkalov, and Iosif Meyerov, editors, *Mathematical Modeling and Supercomputer Technologies*, pages 215–226, Springer, Cham, 2022.
35. Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, January 1996.
36. Andreas M. Tillmann, Daniel Bienstock, Andrea Lodi, and Alexandra Schwartz. Cardinality minimization, constraints, and regularization: A survey. *SIAM Review*, 66(3):403–477, May 2024.
37. Duo Wang, Zheng–Fen Jin, and Youlin Shang. A penalty decomposition method for nuclear norm minimization with ℓ_1 norm fidelity term. *Evolution Equations and Control Theory*, 8(4):695–708, June 2019.
38. Jinming Wen, Changhao Li, Qianyu Shu, and Zhengchun Zhou. Randomized orthogonal matching pursuit algorithm with adaptive partial selection for sparse signal recovery. *SIAM Journal on Imaging Sciences*, 18(2):1028–1057, April 2025.
39. Yong Zhang, Bin Dong, and Zhaosong Lu. ℓ_0 minimization for wavelet frame based image restoration. *Mathematics of Computation*, 82(282):995–1015, August 2013.
40. Yun–Bin Zhao. Optimal k-thresholding algorithms for sparse optimization problems. *SIAM Journal on Optimization*, 30(1):31–55, January 2020.