IOP Publishing Inverse Problems 42 (2026) 055002 https://doi.org/10.1088/1361-6420/ae6477

Inverse Problems

Check for updates

RECEIVED
11 August 2025

REVISED
1 March 2026

ACCEPTED FOR PUBLICATION
24 April 2026

PUBLISHED
11 May 2026

PAPER

General decomposition pursuit algorithm for linear inverse problems

Yun-Bin Zhao[1,*], **Zhong-Feng Sun**[2], **Xin Liu**[3] and **Quan Yu**[4]

[1] Shenzhen International Center for Industrial and Applied Mathematics, SRIBD, The Chinese University of Hong Kong, Shenzhen, People's Republic of China
[2] School of Mathematics and Statistics, Shandong University of Technology, Zibo, Shandong, People's Republic of China
[3] State Key Laboratory of Mathematical Sciences, AMSS, Chinese Academy of Sciences, and University of Chinese Academy of Sciences, Beijing, People's Republic of China
[4] School of Mathematics, Hunan University, Changsha, Hunan, People's Republic of China
[*] Author to whom any correspondence should be addressed.

E-mail: yunbinzhao@cuhk.edu.cn, zfsun@sdut.edu.cn, liuxin@lsec.cc.ac.cn and quanyu@hnu.edu.cn

Keywords: linear inverse problem, sparse solution, general decomposition pursuit, sparsity-aware algorithm, restricted isometry property, signal and image reconstruction

Abstract
Over the past few decades, various numerical methods have been developed to solve linear inverse problems with sparse solutions. However, there remains a shortage of efficient algorithms specifically tailored for large-scale problems. In this paper, we work toward bridging this gap by developing the so-called *General Decomposition Pursuit* algorithm, which is designed to directly tackle a large-scale problem. This algorithmic framework is notably distinct from any existing sparsity-aware method. The main mechanism of the algorithm involves decomposing a large problem into several mutually coupled small subproblems and then combining the (inexact) solutions of these subproblems to generate a sparse solution to the original problem. The global convergence of the algorithm is shown under the restricted isometry property. Simulations with synthetic data and applications in medical image reconstruction indicate that the proposed algorithm, when equipped with an appropriate inner solver for subproblems, can outperform several mainstream algorithms in success rates for locating the sparse solution to the problem.

1. Introduction

Suppose that $x^* \in \mathbb{R}^n$ is an unknown source signal. Let $A \in \mathbb{R}^{m\times n}$ be a given measurement matrix, and let $y := Ax^* + \nu$ be the observed signal, where $\nu \in \mathbb{R}^m$ is a noise vector. Given the data (y, A), the problem of reconstructing the source signal x^* from the measurement equation

$$y = Ax + \nu \tag{1.1}$$

is called a linear inverse problem (LIP) which is a ubiquitous mathematical model arising from many areas of science and engineering (see, e.g. [4, 15, 27, 30, 35, 41, 42]). This problem has a long-lasting impact on system identification, geophysical signal recovery, x-ray tomography, compressive sampling, remote sensing, image deblurring, statistical learning, wireless channel information estimation, to name but a few [4, 9, 19, 26, 28, 31, 36, 41].

Throughout this paper, we assume $m \ll n$, i.e. the number of observations is significantly smaller than the signal length. This assumption reflects a common scenario for ill-posed LIPs arising from signal reconstruction via compressive sampling (see, e.g. [10, 19, 21, 27, 30, 36, 53]). In this situation, reconstructing a signal from (1.1) is generally impossible unless some additional information about the signal is available. Fortunately, many natural signals or transformed signals often exhibit sparse structures. Sparsity serves as a piece of valuable additional information that can be used to facilitate signal reconstruction [9, 19, 36, 42]. In fact, sparsity has become a powerful regularization technique for ill-posed problems [29, 42, 46]. This explains why sparsity-aware algorithms often perform exceptionally well in signal and image processing [19, 36, 42]. Existing sparsity-aware algorithms can be broadly classified into three categories.

Hard thresholding methods. The hard thresholding methods constitute a large family of sparsity-aware algorithms, including iterative hard thresholding (IHT) [6, 22], hard thresholding pursuit (HTP) [20], compressive sampling matching pursuit (CoSaMP) [38], subspace pursuit (SP) [16], Newton-step-based method [39], and heavy-ball-enhanced pseudo-inverse-based method [51]. Recently, the relaxed optimal thresholding pursuit (ROTP) and its modifications were also proposed (see [43, 44, 54, 58]).

Heuristic methods. The orthogonal matching pursuit (OMP) [37, 40, 49] is a typical heuristic method initially developed in statistics and later applied to signal processing. Several modifications of OMP have been proposed in the literature, including weak OMP [45], general OMP [50], and stagewise OMP [18]. The main differences between these methods lie in the basis-selection rules at each iteration.

Convex optimization methods. LASSO [28], ℓ_1-minimization [10, 13, 17], weighted ℓ_1-minimization [11, 53], and Dantzig selector [12, 19] are sparsity-aware convex optimization models. Any method for solving such models qualifies as a convex optimization method. The well-known ISTA and FISTA [4], developed for LASSO, belong to this category, while they are also interpretable as thresholding methods. Their eigenvalue-free variants [48] arise from integration with the majorization–minimization framework. Additionally, reweighted ℓ_1-minimization [11, 53, 56] and dual-density-based methods [55] are also convex optimization approaches for LIPs. However, these methods generally exhibit higher computational complexity than hard-thresholding and heuristic approaches.

It is worth noting that learning-based sparsity-aware algorithms for LIPs have been recently studied in the literature as well (e.g. [3, 5, 7, 14]). However, this topic is beyond the scope of this paper.

Existing methods face significant challenges when solving large-scale problems. Interior-point algorithms for convex optimization require solving dense normal equations at each iteration. OMP necessitates a number of iterations equal to the sparsity level of solution, making it inefficient for high-sparsity problems. Empirical evidence indicates that IHT, HTP, CoSaMP, SP, ISTA, and FISTA generally exhibit lower success rates in locating sparse solutions than OMP and reweighted ℓ_1-minimization methods. Moreover, IHT, ISTA, and FISTA typically require numerous iterations to construct quality solutions. The ROTP algorithm involves solving computationally expensive quadratic optimization subproblems at every iteration; while certain modifications of ROTP can mitigate this cost, the computational burden remains substantial. Thus there is an urgent need for a new generation of sparsity-aware algorithms capable of efficiently handling large-scale LIPs.

In this paper, we propose a novel algorithm termed *general decomposition pursuit* (GDP), specifically tailored for large-scale LIPs. The main idea is to decompose a large problem into a series of coupled small subproblems, which can be solved efficiently using any existing sparsity-aware solver. Roughly speaking, GDP consists of three main steps. (i) The matrix A is divided into p smaller submatrices $A_i, i = 1, \ldots, p$, and y is partitioned into $y = y_1 + \cdots + y_p$, resulting in p subproblems of the form $y_i = A_i z + \nu_i, i = 1, \ldots, p$. (ii) The vectors y_i are adaptively updated in an alternating manner. (iii) The subproblems are solved, and their solutions are merged to form a vector on which hard thresholding and orthogonal projection are applied to generate an approximate solution to the problem. GDP incorporates some features of the alternating direction method of multipliers (ADMM) and distributed optimization algorithms (see, e.g. [2, 8, 23, 24, 32–34]).

The main contributions of this paper are threefold. First, we introduce GDP—a framework of decomposition algorithms for large-scale LIPs (see section 3 for details). GDP provides a novel mechanism for ensembling existing sparsity-aware methods as subproblem solvers to tackle a large-scale problem. Second, we establish global convergence of GDP under the restricted isometry property (RIP), a standard assumption in the compressive-sampling literature. Theoretical guarantees for both noiseless and noisy environments are presented in theorem 4.5 and corollary 4.6, respectively. These are the first results of their kind for decomposition-type methods. Finally, numerical experiments demonstrate that GDP achieves superior signal reconstruction quality and higher success rates in locating true sparse solutions of LIPs than existing methods (see section 5 for details).

2. Preliminaries

Notation: Throughout the paper, all vectors are understood as column vectors unless otherwise specified. For $v = (v_1^T, \ldots, v_p^T)^T$, where each v_i is a vector, we write it as $v = (v_1, \ldots, v_p)$ when no confusion arises.

For $S \subseteq \{1,\ldots,n\}$, $|S|$ denotes the cardinality of S, and $\overline{S} = \{1,\ldots,n\} \setminus S$ denotes the complement of S. Given $x = (x_1,\ldots,x_n)^T \in \mathbb{R}^n$, the index set $\operatorname{supp}(x) = \{i : x_i \neq 0\}$ is called the support of x. x is said to be K-sparse if the cardinality of its support is less than or equal to K, i.e. $|\operatorname{supp}(x)| \leqslant K$. Given $x \in \mathbb{R}^n$ and $S \subseteq \{1,\ldots,n\}$, x_S denotes the vector obtained by keeping x_i for $i \in S$ and setting $x_i = 0$ for $i \notin S$. (In this case, $x_S \in \mathbb{R}^n$.) We also use x_S to denote the compressed vector by deleting the entries x_i for $i \notin S$ from x. (In this case, $x_S \in \mathbb{R}^{|S|}$.) The meaning of x_S is clear from the context. Given $S \subseteq \{1,\ldots,n\}$ and $A \in \mathbb{R}^{m\times n}$, A_S denotes the submatrix with columns indexed by S. Sorting the absolute entries of x into descending order $|x_{i_1}| \geqslant |x_{i_2}| \geqslant \cdots \geqslant |x_{i_n}|$, let $\mathcal{L}_K(x) = \{i_1, i_2, \ldots, i_K\}$ be the index set for the K largest-magnitude entries of x. Let $\mathcal{H}_K(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ be the hard thresholding operator, which retains the K largest-magnitude entries of a vector and sets the other entries to zero. Let $\sigma_K(x) = \|x - \mathcal{H}_K(x)\|_2$ denote the residual of approximation of x via $\mathcal{H}_K(x)$.

Definition 2.1. For the vector $u = (u_1,\ldots,u_p) \in \mathbb{R}^n$, where $u_i \in \mathbb{R}^{n_i}$ for $i = 1,\ldots,p$ and $n_1 + \cdots + n_p = n$, the norm $\|\cdot\|_{(p,\infty)} : \mathbb{R}^n \to \mathbb{R}_+$ is defined as

$$\|u\|_{(p,\infty)} = \max_{1\leqslant i\leqslant p} \|u_i\|_2.$$

The following result (see [57, lemma 3.2]) is very useful in the analysis of sparsity-aware algorithms using $\mathcal{H}_K(\cdot)$.

Lemma 2.2 [57]. *For any vector $z \in \mathbb{R}^n$ and any K-sparse vector $x \in \mathbb{R}^n$, one has $\|x - \mathcal{H}_K(z)\|_2 \leqslant \frac{\sqrt{5}+1}{2} \|(x-z)_{S\cup T}\|_2$, where $S = \operatorname{supp}(x)$ and $T = \operatorname{supp}(\mathcal{H}_K(z))$.*

The definition below was first introduced in [10, definition 1.1]. It plays a vital role in the analysis of our algorithm.

Definition 2.3 [10]. A matrix $A \in \mathbb{R}^{m\times n} (m \ll n)$ is said to satisfy the RIP of order t, where $t \in \{1,\ldots,m\}$, if there exists a smallest constant $0 \leqslant \delta_t < 1$ such that

$$(1-\delta_t)\|x\|_2^2 \leqslant \|Ax\|_2^2 \leqslant (1+\delta_t)\|x\|_2^2 \tag{2.1}$$

for any t-sparse vector x. δ_t is referred to as the restricted isometry constant (RIC) of order t.

Inequality (2.1) implies that any t columns of A are linearly independent, and hence $t \leqslant m$. The next lemma (see [21, proposition 6.3]) follows directly from definition 2.3.

Lemma 2.4 [21]. *Assume that the matrix $A \in \mathbb{R}^{m\times n}$ satisfies the RIP. Let $u, v \in \mathbb{R}^n$ be s-sparse and t-sparse vectors, respectively. If $\operatorname{supp}(u) \cap \operatorname{supp}(v) = \emptyset$, then*

$$|u^T A^T A v| \leqslant \delta_{s+t} \|u\|_2 \|v\|_2.$$

The following lemma is obtained as a special case of lemma 4.2 in [57].

Lemma 2.5 [57]. *Let $y := Ax^* + \nu$ be the observations of the K-sparse vector x^*, where ν denotes the observation error. Let $\Lambda \subseteq \{1,\ldots,n\}$, $\overline{\Lambda} = \{1,\ldots,n\} \setminus \Lambda$, and $\widehat{z}$ be the solution to the orthogonal projection*

$$\widehat{z} = \arg\min_{z\in\mathbb{R}^n} \{\|y - Az\|_2 : \operatorname{supp}(z) \subseteq \Lambda\}.$$

If the matrix A satisfies the RIP of order $|\Lambda| + K$, i.e. $\delta_{|\Lambda|+K} < 1$, then

$$\|\widehat{z} - x^*\|_2 \leqslant \frac{1}{\sqrt{1-\delta_{|\Lambda|+K}^2}} \|(\widehat{z} - x^*)_{\overline{\Lambda}}\|_2 + \frac{\|A^T\nu\|_2}{1-\delta_{|\Lambda|+K}}.$$

3. GDP

We first outline the main idea of the method. For simplicity, suppose that A is a full-row-rank matrix. Let A_Γ with $|\Gamma| = m$ be a nonsingular square submatrix of A. Let $u \in \mathbb{R}^n$ be the vector with $u_{\overline{\Gamma}} = 0$ and $u_\Gamma = A_\Gamma^{-1}\nu$. Thus $\nu = A_\Gamma u_\Gamma + A_{\overline{\Gamma}} u_{\overline{\Gamma}} = Au$. We split A into p blocks, i.e. $A = [A_1,\ldots,A_p]$, where $A_i \in \mathbb{R}^{m\times n_i}$ with $m \ll n_i$ and $n_1 + \cdots + n_p = n$. Let x^* and u be split accordingly as $x^* = (x_1^*,\ldots,x_p^*)$ and $u = (u_1,\ldots,u_p)$, where $x_i^*, u_i \in \mathbb{R}^{n_i}$ for $i = 1,\ldots,p$. Then

$$y = Ax^* + \nu = A(x^* + u) = A_1(x_1^* + u_1) + \cdots + A_p(x_p^* + u_p) = y_1 + \cdots + y_p,$$

where $y_i = A_i(x_i^* + u_i) = A_i x_i^* + \nu_i$ with $\nu_i = A_i u_i$ and $\nu_1 + \cdots + \nu_p = \nu$. We call the vectors y_i the optimal allocations of the observation y between p subproblems. If such allocations are available and if one can solve the subproblems

$$y_i = A_i z + \nu_i, \ i = 1, \ldots, p \tag{3.1}$$

to obtain the solutions $z = x_i^* \in \mathbb{R}^{n_i}$, $i = 1, \ldots, p$, then x^* can be immediately reconstructed by merging the solutions of (3.1). This motivates us to develop a novel algorithm to iteratively search for the optimal allocations $y_1, \ldots, y_p$. To minimize the cost per iteration, we may update the allocation in an alternating manner, focusing on one subproblem at a time while utilizing the current approximate allocations for the other subproblems.

Specifically, suppose that $x^{(k)} \in \mathbb{R}^n$ is the current iterate and that $y_1^{(k)}, \ldots, y_p^{(k)}$ are the current allocations between subproblems. The idea of our algorithm can be outlined as follows. Begin with the first subproblem determined by A_1. Given $y_2^{(k)}, \ldots, y_p^{(k)}$, let

$$\phi_1^{(k)} = y - \left(y_2^{(k)} + \cdots + y_p^{(k)}\right)$$

be the initial allocation to this subproblem. Use any existing sparsity solver to find a sparse solution to the subproblem $\phi_1^{(k)} = A_1 z_1$, and denote the resulting sparse solution by z_1. Performing hard thresholding on z_1 yields $\widetilde{x}_1^{(k)} = \mathcal{H}_K(z_1)$. Then $\widetilde{y}_1^{(k)} := A_1 \widetilde{x}_1^{(k)}$ is the updated allocation for the first subproblem. Next, let

$$\phi_2^{(k)} := y - \widetilde{y}_1^{(k)} - \left(y_3^{(k)} + \cdots + y_p^{(k)}\right)$$

be the initial allocation for the second subproblem. Solve the subproblem $\phi_2^{(k)} = A_2 z_2$ by any sparsity-aware solver to get a sparse vector z_2, and set $\widetilde{x}_2^{(k)} = \mathcal{H}_K(z_2)$. Then $\widetilde{y}_2^{(k)} := A_2 \widetilde{x}_2^{(k)}$ is the update of $y_2^{(k)}$. Continue by taking

$$\phi_i^{(k)} := y - \left(\widetilde{y}_1^{(k)} + \cdots + \widetilde{y}_{i-1}^{(k)}\right) - \left(y_{i+1}^{(k)} + \cdots + y_p^{(k)}\right)$$

as the initial allocation for the ith subproblem in order to generate $\widetilde{y}_i^{(k)}$, the update of $y_i^{(k)}$. This process continues until $\widetilde{x}_p^{(k)}$ is obtained. Merging all K-sparse vectors $\widetilde{x}_i^{(k)}$ yields the aggregated one $\widetilde{x}^{(k)} = (\widetilde{x}_1^{(k)}, \ldots, \widetilde{x}_p^{(k)})$, which is not necessarily K-sparse. Similar to CoSaMP and SP, we may perform a shrinkage of $\widetilde{x}^{(k)}$, followed by a pursuit step (i.e. orthogonal projection), to generate the next iterate $x^{(k+1)}$.

We use an existing sparsity-aware algorithm to solve the subproblems $\phi_i^{(k)} = A_i z_i$ for $i = 1, \ldots, p$, which are significantly smaller in scale than the original problem. Existing algorithms that can serve this purpose include OMP, HTP, CoSaMP, SP, ROTP, and ℓ_1-minimization. Throughout the rest of the paper, we use SPARSITY-SOLVER$(\widehat{y}, \widehat{B})$ to denote such a solver, where $\widehat{y} \in \mathbb{R}^m$ is a vector and $\widehat{B} \in \mathbb{R}^{m \times \ell}$ is a full-row-rank matrix with $m \ll \ell$. The solver takes the problem data $(\widehat{y}, \widehat{B})$ to generate a sparse solution (denoted by $\widehat{x}$) for the system $\widehat{y} = \widehat{B} z$:

$$\widehat{x} := \arg\left[\text{SPARSITY-SOLVER}\left(\widehat{y}, \widehat{B}\right)\right].$$

We now formally describe the algorithm, referred to as the GDP in this paper.

GDP. Input $A = [A_1, \ldots, A_p] \in \mathbb{R}^{m \times n}$ ($A_i \in \mathbb{R}^{m \times n_i}, i = 1, \ldots, p$ with $m \ll \min_{1 \leqslant i \leqslant p} n_i$), $y \in \mathbb{R}^m$, sparsity level K, tolerance $\epsilon > 0$, and integer τ satisfying $K \leqslant \tau \leqslant 2K$. Set $k := 0$ (iteration counter).

S1 Let $z_i^{(0)} = \arg[\text{SPARSITY-SOLVER}(y, A_i)]$, $x_i^{(0)} = \mathcal{H}_K(z_i^{(0)})$ and $y_i^{(0)} = A_i x_i^{(0)}$ for $i = 1, \ldots, p$. If $\sigma_K(z_i^{(0)}) \leqslant \epsilon$ for some $i \in \{1, \ldots, p\}$, stop and return $\widehat{x} = (0, \ldots, 0, x_i^{(0)}, 0, \ldots, 0) \in \mathbb{R}^n$.

S2 Given the current iterate $x^{(k)}$ and $y_i^{(k)}$ for $i = 2, \ldots, p$, set $\phi_1^{(k)} := y - \sum_{i=2}^p y_i^{(k)}$ and perform the loop
- for $i = 1, \ldots, p-1$, do
  - $z_i^{(k)} = \arg[\text{SPARSITY-SOLVER}(\phi_i^{(k)}, A_i)]$; $\widetilde{x}_i^{(k)} = \mathcal{H}_K(z_i^{(k)})$;
  - $\widetilde{y}_i^{(k)} = A_i \widetilde{x}_i{}^{(k)}$;
  - $\phi_{i+1}^{(k)} = y - \sum_{j=1}^i \widetilde{y}_j^{(k)} - \sum_{j=i+2}^p y_j^{(k)}$;

- end
  - $z_p^{(k)} = \arg[\text{SPARSITY-SOLVER}(\phi_p^{(k)}, A_p)]$; $\widetilde{x}_p^{(k)} = \mathcal{H}_K(z_p^{(k)})$.

S3 Let $d^{(k)} = A^T(y - Ax^{(k)})$, $\widetilde{x}^{(k)} = (\widetilde{x}_1^{(k)}, \ldots, \widetilde{x}_p^{(k)})$ and $T^{(k)} = \mathcal{L}_\tau(\widetilde{x}^{(k)}) \cup \mathcal{L}_{2K-\tau}(d^{(k)})$. Set

$$\widehat{x}^{(k)} = \arg\min_{x\in\mathbb{R}^n} \left\{ \|y - Ax\|_2 : \ \text{supp}\,(x) \subseteq T^{(k)} \right\}. \tag{3.2}$$

S4 Let $S^{(k)} = \mathcal{L}_K(\widehat{x}^{(k)})$ and

$$\left(x_1^{(k+1)}, \ldots, x_p^{(k+1)}\right) = x^{(k+1)} := \arg\min_{x\in\mathbb{R}^n} \left\{ \|y - Ax\|_2 : \ \text{supp}\,(x) \subseteq S^{(k)} \right\}, \tag{3.3}$$

where $x_i^{(k+1)} \in \mathbb{R}^{n_i}, i = 1, \ldots, p$. Set $y_i^{(k+1)} = A_i x_i^{(k+1)}$ for $i = 2, \ldots, p$. Replace $k+1$ by k and repeat S2-S4 until a stopping criterion is met.

The SPARSITY-SOLVER$(\cdot,\cdot)$ is referred to as the *inner solver* of the GDP algorithm. The algorithm can start from any initial point $x^{(0)} \in \mathbb{R}^n$. However, we recommend using the initial point generated at S1, as this might terminate the algorithm early if an ϵ-solution is discovered there, which avoids unnecessary iterations. At S2, the GDP employs an alternating strategy to generate an intermediate point $\widetilde{x}^{(k)}$, which is used to produce $x^{(k+1)}$ through the pursuit steps S3 and S4. Two consecutive shrinkages are performed: The first one in S3, which is relatively loose with cardinality $|T^{(k)}| \leqslant 2K$, significantly reduces the error metric $\|y - Ax\|_2$; the second in S4 ensures that the iterate is K-sparse. This alternating scheme is inspired by the traditional Gauss–Seidel iterative framework for solving linear equations and is also used in ADMM and distributed optimization algorithms (see, e.g. [2, 8, 23, 24, 32–34]). The stopping criterion at S4 can be tailored to specific applications. For example, we can stop the algorithm when the prescribed maximum number of iterations is reached or when the point $x^{(k+1)}$ satisfies $\|y - Ax^{(k+1)}\|_2 \leqslant \epsilon$. The algorithm is termed *general* due to its flexible framework, allowing for different choices of inner solvers (exact or inexact), different selections of gradient information, and different choices for the number and size of subproblems.

Remark 3.1. When GDP is used to solve a problem, the number of blocks p adopted in the algorithm depends on the actual size of the problem and the size of subproblems that can be handled by the user's available computing resources. Clearly, $p = 2$ and $p = 3$ are two particularly important special cases of GDP. For these cases, the algorithm is termed GDP_2 and GDP_3, respectively. The performance of GDP_2 and GDP_3 is evaluated and discussed in section 5. In addition, while the GDP in its current version does not seem suitable for parallel computing due to coupled subsystems, the framework of GDP under suitable modification could be parallelized. For instance, if the Gauss–Seidel-type iteration is replaced with a Jacobi-type iteration in S2, then the term $\phi_i^{(k)}$ in S2 can be updated as $\phi_i^{(k)} = y - \sum_{j\neq i} y_j^{(k)}$ for $i = 1, \ldots, p$ and thus the subproblems $\phi_i^{(k)} = A_i z_i$ for $i = 1, \ldots, p$ can be solved in parallel.

4. Analysis of algorithm

The GDP algorithm provides a way to leverage algorithms for relatively small problems to tackle larger ones. GDP allows the subproblems to be solved inexactly using any sparsity-aware method. Without loss of generality, we make the following assumption on the solver for subproblems:

Assumption 4.1. Given $\epsilon > 0, \widehat{y} \in \mathbb{R}^m$ and full-row-rank matrix $\widehat{B} \in \mathbb{R}^{m\times\ell}$ with $m \ll \ell$, the solution

$$\widehat{z} = \arg\left[\text{SPARSITY-SOLVER}\left(\widehat{y}, \widehat{B}\right)\right]$$

satisfies $|\text{supp}(\widehat{z})| \leqslant m$ and $\|\widehat{y} - \widehat{B}\widehat{z}\|_2 \leqslant \epsilon$.

This assumption is generally satisfied by any existing efficient sparsity-aware algorithm. In fact, since $\widehat{B}$ is a full-row-rank matrix, there exists a solution $\bar{z}$ to the system $\widehat{y} = \widehat{B}z$, which satisfies $|\text{supp}(\bar{z})| \leqslant m$. Thus by the continuity of the linear mapping $z \mapsto \widehat{B}z$, for any $\epsilon > 0$, the linear system $\widehat{y} = \widehat{B}z$ always has an ϵ-solution $\widehat{z}$ with $\text{supp}(\widehat{z}) \subseteq \text{supp}(\bar{z})$, satisfying $|\text{supp}(\widehat{z})| \leqslant m$ and $\|\widehat{y} - \widehat{B}\widehat{z}\|_2 \leqslant \epsilon$. Any existing numerical methods for locating a solution to underdetermined linear systems can be used to generate such an approximate solution satisfying assumption 4.1. The benefit of using a SPARSITY-SOLVER$(\widehat{y}, \widehat{B})$

is that when the underlying linear system has a sparse solution, the sparsity level of the computed $\widehat{z}$ is usually much lower than m.

4.1. Analysis of initial step

Before establishing the main result for GDP, we first justify the initial step of GDP by showing that if $\sigma_K(z_i^{(0)}) \leqslant \epsilon$ for some i, then the algorithm immediately terminates at a K-sparse ϵ-solution to the problem and hence no further iteration is required.

Theorem 4.2. *Let $A = [A_1, \ldots, A_p] \in \mathbb{R}^{m\times n}$, where every $A_i \in \mathbb{R}^{m\times n_i}$ with $m \ll n_i$ is a full-row-rank matrix. Suppose that $x^* \in \mathbb{R}^n$ is K-sparse, and that $y := Ax^* + \nu$ are the observations with noise $\nu \in \mathbb{R}^m$. Assume that the RIC of order $2K$ of A satisfies $\delta_{2K} < 1$ (this implies $2K \leqslant m$). Let $z_i^{(0)}$ and $x_i^{(0)}$ be defined at S1 of GDP with inner solvers obeying assumption 4.1. If $\sigma_K(z_i^{(0)}) \leqslant \epsilon$ for some $i \in \{1, \ldots, p\}$, then the point $\widehat{x} = (0, \ldots, 0, x_i^{(0)}, 0, \ldots, 0) \in \mathbb{R}^n$ satisfies that*

$$\|\widehat{x} - x^*\|_2 \leqslant \frac{1}{\sqrt{1-\delta_{2K}}}\|\nu\|_2 + C^*\left(\|A\|_2 + 1\right)\left(\frac{\|A\|_2}{\sqrt{1-\delta_{2K}}} + 1\right)\epsilon, \tag{4.1}$$

where C^* is a constant determined by A. Moreover, if the observations are accurate (i.e. $\nu = 0$) and $\sigma_K(z_i^{(0)}) = 0$ for some $i \in \{1, \ldots, p\}$, then $\widehat{x} = x^*$.

Proof. Let $z_i^{(0)}$ be the output of SPARSITY-SOLVER(y, A_i) in S1 of GDP. By assumption 4.1, we have $\|y - A_i z_i^{(0)}\|_2 \leqslant \epsilon$. Notice that

$$y - A_i x_i^{(0)} = y - A_i \mathcal{H}_K\left(z_i^{(0)}\right) = y - A_i z_i^{(0)} - A_i\left(\mathcal{H}_K\left(z_i^{(0)}\right) - z_i^{(0)}\right).$$

Suppose that $\sigma_K(z_i^{(0)}) \leqslant \epsilon$, which implies $\|\mathcal{H}_K(z_i^{(0)}) - z_i^{(0)}\|_2 \leqslant \epsilon$. Then,

$$\|y - A_i x_i^{(0)}\|_2 \leqslant \|y - A_i z_i^{(0)}\|_2 + \|A_i\left(\mathcal{H}_K\left(z_i^{(0)}\right) - z_i^{(0)}\right)\|_2 \leqslant (\|A\|_2 + 1)\,\epsilon, \tag{4.2}$$

where the last inequality follows form $\|A_i\|_2 \leqslant \|A\|_2$. Note that every A_j is a full-row-rank matrix. Let B_j be an $m \times m$ nonsingular submatrix of A_j. For every $j \neq i$, define the vector $v_j = (B_j)^{-1}(y - A_i x_i^{(0)})$. Then for every $j \neq i$, by using (4.2), we have that

$$\|v_j\|_2 = \|\left(B_j\right)^{-1}\left(y - A_i x_i^{(0)}\right)\|_2 \leqslant \|\left(B_j\right)^{-1}\|_2 \|y - A_i x_i^{(0)}\|_2 \leqslant C^*\left(\|A\|_2 + 1\right)\epsilon, \tag{4.3}$$

where C^* is the constant given as

$$C^* = \max_{1\leqslant j\leqslant p} \max_{B} \left\{\|B^{-1}\|_2 : B \in \mathbb{R}^{m\times m} \text{ is a nonsingular submatrix of } A_j\right\}.$$

By the definition of v_j, we see that $B_j v_j = y - A_i x_i^{(0)}$ for every $j \neq i$. Adding up yields $\sum_{j\neq i} B_j v_j = (p-1)(y - A_i x_i^{(0)})$, which implies that

$$y = A_i x_i^{(0)} + \sum_{j\neq i} B_j\left(\frac{v_j}{p-1}\right) = A_i x_i^{(0)} + \sum_{j\neq i} A_j \widetilde{v}_j, \tag{4.4}$$

where $\widetilde{v}_j \in \mathbb{R}^{n_j}$ is an inflated vector of $v_j/(p-1)$ by adding $n_j - m$ zeros as entries. Notice that

$$y = Ax^* + \nu = A_i x_i^* + \sum_{j\neq i} A_j x_j^* + \nu.$$

It follows from (4.4) that

$$A_i\left(x_i^{(0)} - x_i^*\right) + \sum_{j\neq i} A_j\left(\widetilde{v}_j - x_j^*\right) = \nu. \tag{4.5}$$

Denote by $\Lambda = \mathrm{supp}(x_i^{(0)})$ and $S_j = \mathrm{supp}(x_j^*)$for $j = 1 \ldots, p$. For $j \neq i$, one has

$$A_j\left(\widetilde{v}_j - x_j^*\right) = \left(A_j\right)_{S_j}\left(\widetilde{v}_j - x_j^*\right)_{S_j} + \left(A_j\right)_{\overline{S_j}}\left(\widetilde{v}_j\right)_{\overline{S_j}}, \tag{4.6}$$

where $\overline{S_j}$ is the complement of S_j with respect to the index set for the columns of A_j. Since $\mathrm{supp}(x_i^{(0)} - x_i^*) \subseteq \Lambda \cup S_i$, we have $A_i(x_i^{(0)} - x_i^*) = (A_i)_{\Lambda\cup S_i}(x_i^{(0)} - x_i^*)_{\Lambda\cup S_i}$. Thus combing (4.5) and (4.6) leads to

$$(A_i)_{\Lambda\cup S_i}\left(x_i^{(0)} - x_i^*\right)_{\Lambda\cup S_i} + \sum_{j\neq i}(A_j)_{S_j}\left(\widetilde{v}_j - x_j^*\right)_{S_j} = \nu - \sum_{j\neq i}(A_j)_{\overline{S_j}}(\widetilde{v}_j)_{\overline{S_j}}. \tag{4.7}$$

Let $Q(\Lambda)$ denote the coefficient matrix on the left-hand side of (4.7), i.e.

$$Q(\Lambda) = \left[(A_i)_{\Lambda\cup S_i}, (A_1)_{S_1}, \ldots, (A_{i-1})_{S_{i-1}}, (A_{i+1})_{S_{i+1}}, \ldots, (A_p)_{S_p}\right].$$

Then the system (4.7) can be written as

$$Q(\Lambda)\begin{pmatrix} \left(x_i^{(0)} - x_i^*\right)_{\Lambda\cup S_i} \\ (\widetilde{v}_1 - x_1^*)_{S_1} \\ \vdots \\ (\widetilde{v}_{i-1} - x_{i-1}^*)_{S_{i-1}} \\ (\widetilde{v}_{i+1} - x_{i+1}^*)_{S_{i+1}} \\ \vdots \\ (\widetilde{v}_p - x_p^*)_{S_p} \end{pmatrix} = \nu - \sum_{j\neq i}(A_j)_{\overline{S_j}}(\widetilde{v}_j)_{\overline{S_j}}.$$

Note that the number of columns of $Q(\Lambda)$ is less than or equal to $|\Lambda| + \sum_{\ell=1}^{p}|S_\ell| \leqslant 2K$ due to $|\Lambda| \leqslant K$ and $\sum_{\ell=1}^{p}|S_\ell| = |\mathrm{supp}(x^*)| \leqslant K$. Thus taking the ℓ_2-norm in both sides of the above equation and using the RIC constant δ_{2K} yields

$$\sqrt{1-\delta_{2K}}\left\|\begin{pmatrix} \left(x_i^{(0)} - x_i^*\right)_{\Lambda\cup S_i} \\ (\widetilde{v}_1 - x_1^*)_{S_1} \\ \vdots \\ (\widetilde{v}_{i-1} - x_{i-1}^*)_{S_{i-1}} \\ (\widetilde{v}_{i+1} - x_{i+1}^*)_{S_{i+1}} \\ \vdots \\ (\widetilde{v}_p - x_p^*)_{S_p} \end{pmatrix}\right\|_2 \leqslant \|\nu\|_2 + \|\sum_{j\neq i}(A_j)_{\overline{S_j}}(\widetilde{v}_j)_{\overline{S_j}}\|_2. \tag{4.8}$$

The last term on the right-hand side of (4.8) can be bounded as

$$\|\sum_{j\neq i}(A_j)_{\overline{S_j}}(\widetilde{v}_j)_{\overline{S_j}}\|_2 \leqslant \sum_{j\neq i}\|(A_i)_{\overline{S_j}}\|_2\|(\widetilde{v}_j)_{\overline{S_j}}\|_2 \leqslant C^*\|A\|_2(\|A\|_2+1)\epsilon, \tag{4.9}$$

where the last inequality follows from $\|(A_i)_{\overline{S_j}}\|_2 \leqslant \|A\|_2$ and (4.3), which implies that $\|(\widetilde{v}_j)_{\overline{S_j}}\|_2 \leqslant \|\widetilde{v}_j\|_2 = \frac{1}{p-1}\|v_j\|_2 \leqslant \frac{1}{p-1}C^*(\|A\|_2+1)\epsilon$. Thus $\sum_{j\neq i}\|\widetilde{v}_j\|_2 \leqslant C^*(\|A\|_2+1)\epsilon$. By the definition of $\widehat{x}$, we have

$$\|\widehat{x} - x^*\|_2 \leqslant \left\|\begin{pmatrix} \left(x_i^{(0)} - x_i^*\right)_{\Lambda\cup S_i} \\ (-x_1^*)_{S_1} \\ \vdots \\ (-x_{i-1}^*)_{S_{i-1}} \\ (-x_{i+1}^*)_{S_{i+1}} \\ \vdots \\ (-x_p^*)_{S_p} \end{pmatrix}\right\|_2 = \left\|\begin{pmatrix} \left(x_i^{(0)} - x_i^*\right)_{\Lambda\cup S_i} \\ (\widetilde{v}_1 - x_1^*)_{S_1} - (\widetilde{v}_1)_{S_1} \\ \vdots \\ (\widetilde{v}_{i-1} - x_{i-1}^*)_{S_{i-1}} - (\widetilde{v}_{i-1})_{S_{i-1}} \\ (\widetilde{v}_{i+1} - x_{i+1}^*)_{S_{i+1}} - (\widetilde{v}_{i+1})_{S_{i+1}} \\ \vdots \\ (\widetilde{v}_p - x_p^*)_{S_p} - (\widetilde{v}_p)_{S_p} \end{pmatrix}\right\|_2.$$

Thus, we have

$$\|\widehat{x}-x^*\|_2 \leqslant \left\| \begin{pmatrix} \left(x_i^{(0)}-x_i^*\right)_{\Lambda\cup S_i} \\ (\widetilde{v}_1-x_1^*)_{S_1} \\ \vdots \\ (\widetilde{v}_{i-1}-x_{i-1}^*)_{S_{i-1}} \\ (\widetilde{v}_{i+1}-x_{i+1}^*)_{S_{i+1}} \\ \vdots \\ (\widetilde{v}_p-x_p^*)_{S_p} \end{pmatrix} \right\|_2 + \sum_{j\neq i}\|\widetilde{v}_j\|_2$$
$$\leqslant \frac{\|\nu\|_2 + C^*\|A\|_2(\|A\|_2+1)\epsilon}{\sqrt{1-\delta_{2K}}} + C^*(\|A\|_2+1)\epsilon, \tag{4.10}$$

where the last inequality follows from (4.8) and (4.9). The bound in (4.10) is exactly the one in (4.1). Moreover, if $\nu=0$ and $\epsilon=0$, then $\widehat{x}=x^*$. □

4.2. Convergence of GDP

In general, the algorithm does not terminate at the initial step and thus generates a sequence $\{x^{(k)}\}$. We now establish the global convergence of this sequence under suitable conditions. Before proving the main result, we first establish two technical lemmas. The first lemma bounds $\|x^{(k+1)}-x^*\|_2$ in terms of $\|\widetilde{x}^{(k)}-x^*\|_{(p,\infty)}$, where $\widetilde{x}^{(k)}$ is defined in GDP.

Lemma 4.3. *Let $x^*\in\mathbb{R}^n$ be a K-sparse vector, and let $y:=Ax^*+\nu$ be the given observations with noise $\nu\in\mathbb{R}^m$. Suppose that A satisfies $\delta_{3K}<1$ (and hence $3K\leqslant m$). Then the sequences $\{\widetilde{x}^{(k)}\}_{k\geqslant 1}$ and $\{x^{(k+1)}\}_{k\geqslant 1}$ generated by the GDP satisfy that*

$$\|x^{(k+1)}-x^*\|_2 \leqslant \omega^2\rho_1\rho_2\sqrt{p}\|\widetilde{x}^{(k)}-x^*\|_{(p,\infty)} + \left(\frac{\omega\rho_2}{1-\delta_{3K}} + \frac{1}{1-\delta_{2K}}\right)\|A^T\nu\|_2, \tag{4.11}$$

where $\omega=(\sqrt{5}+1)/2,\ \rho_1=\frac{1}{\sqrt{1-\delta_{3K}^2}}$ *and* $\rho_2=\frac{1}{\sqrt{1-\delta_{2K}^2}}$.

Proof. Denote by $\Lambda=\mathcal{L}_\tau(\widetilde{x}^{(k)})$ and $g:=\mathcal{H}_\tau(\widetilde{x}^{(k)})=(\widetilde{x}^{(k)})_\Lambda$. Clearly, $\mathrm{supp}(g)\subseteq\Lambda$. By lemma 2.2, we have

$$\|g-x^*\|_2 \leqslant \omega\left\|\left(\widetilde{x}^{(k)}-x^*\right)_{S\cup\Lambda}\right\|_2 \leqslant \omega\|\widetilde{x}^{(k)}-x^*\|_2 \leqslant \omega\sqrt{p}\|\widetilde{x}^{(k)}-x^*\|_{(p,\infty)}, \tag{4.12}$$

where $S=\mathrm{supp}(x^*)$. According to S3 of GDP, $T^{(k)}=\mathcal{L}_\tau(\widetilde{x}^{(k)})\cup\mathcal{L}_{2K-\tau}(d^{(k)})$. Notice that $\mathrm{supp}(g)\subseteq\Lambda\subseteq T^{(k)}$ which implies that $g_{\overline{T^{(k)}}}=0$, where $\overline{T^{(k)}}=\{1,\ldots,n\}\setminus T^{(k)}$. By the structure of GDP, the $2K$-sparse vector $\widehat{x}^{(k)}$ is the solution to the least-squares problem in (3.2). By using lemma 2.5, and by noting that $|T^{(k)}|\leqslant 2K$ and $(\widehat{x}^{(k)})_{\overline{T^{(k)}}}=0=g_{\overline{T^{(k)}}}$, we have that

$$\|\widehat{x}^{(k)}-x^*\|_2 \leqslant \frac{1}{\sqrt{1-\left(\delta_{|T^{(k)}|+K}\right)^2}}\left\|\left(\widehat{x}^{(k)}-x^*\right)_{\overline{T^{(k)}}}\right\|_2 + \frac{\|A^T\nu\|_2}{1-\delta_{|T^{(k)}|+K}}$$
$$= \frac{1}{\sqrt{1-(\delta_{3K})^2}}\|(g-x^*)_{\overline{T^{(k)}}}\|_2 + \frac{\|A^T\nu\|_2}{1-\delta_{3K}} \leqslant \rho_1\|g-x^*\|_2 + \frac{\|A^T\nu\|_2}{1-\delta_{3K}},$$

where $\rho_1=\frac{1}{\sqrt{1-\delta_{3K}^2}}$. Let $S^{(k)}=\mathcal{L}_K(\widehat{x}^{(k)})$ and $h=\mathcal{H}_K(\widehat{x}^{(k)})=(\widehat{x}^{(k)})_{S^{(k)}}$. It follows from lemma 2.2 and the above inequality that

$$\|h-x^*\|_2 \leqslant \omega\left\|\left(\widehat{x}^{(k)}-x^*\right)_{S\cup\mathrm{supp}(h)}\right\|_2 \leqslant \omega\rho_1\|g-x^*\|_2 + \frac{\omega\|A^T\nu\|_2}{1-\delta_{3K}}. \tag{4.13}$$

Notice that $|S^{(k)}| = K$ and $h_{\overline{S^{(k)}}} = 0$, where $\overline{S^{(k)}} = \{1,\ldots,n\} \setminus S^{(k)}$. Since $x^{(k+1)} = (x_1^{(k+1)},\ldots,x_p^{(k+1)})$ is the solution to (3.3), by lemma 2.5, one has

$$\begin{aligned}
\|x^{(k+1)} - x^*\|_2 &\leqslant \frac{1}{\sqrt{1-\left(\delta_{|S^{(k)}|+K}\right)^2}} \| \left(x^{(k+1)} - x^*\right)_{\overline{S^{(k)}}} \|_2 + \frac{\|A^T\nu\|_2}{1-\delta_{|S^{(k)}|+K}} \\
&= \frac{1}{\sqrt{1-(\delta_{2K})^2}} \| (h - x^*)_{\overline{S^{(k)}}} \|_2 + \frac{\|A^T\nu\|_2}{1-\delta_{2K}} \\
&\leqslant \omega\rho_1\rho_2 \|g - x^*\|_2 + \left(\frac{\omega\rho_2}{1-\delta_{3K}} + \frac{1}{1-\delta_{2K}}\right)\|A^T\nu\|_2, \qquad (4.14)
\end{aligned}$$

where $\rho_2 = \frac{1}{\sqrt{1-\delta_{2K}^2}}$, the second relation above follows from $(x^{(k+1)})_{\overline{S^{(k)}}} = 0 = h_{\overline{S^{(k)}}}$, and the last inequality follows from (4.13) and $\|(h-x^*)_{\overline{S^{(k)}}}\|_2 \leqslant \|h - x^*\|_2$. Merging (4.12) and (4.14) yields (4.11) immediately. □

We now present the next technical result.

Lemma 4.4. *Let $A = [A_1,\ldots,A_p] \in \mathbb{R}^{m\times n}$, where each $A_i \in \mathbb{R}^{m\times n_i}$ with $m \ll n_i$ and $n_1 + \cdots + n_p = n$. Suppose that A satisfies $\delta_{4K} < 1$ (and hence $4K \leqslant m$). For a given integer $1 \leqslant q \leqslant p$ and a vector $u \in \mathbb{R}^{n_q}$, let $\chi \subseteq \{1,\ldots,n_q\}$ be an index set satisfying $\mathcal{L}_K(u) \subseteq \chi$ and $|\chi| \leqslant 3K$. Then*

$$|u_\chi^T (A_q)_\chi^T (A_q)_{\overline{\chi}} u_{\overline{\chi}}| \leqslant \ell^* \delta_{|\chi|+K} \|u_\chi\|_2^2, \qquad (4.15)$$

where $\overline{\chi} = \{1,\ldots,n_q\} \setminus \chi$ and $\ell^* := \frac{n-m(p-1)}{K} \geqslant \frac{n_q}{K}$.

Proof. Decompose the set $\overline{\chi}$ as

$$\overline{\chi} = \Gamma_1 \cup \cdots \cup \Gamma_{\ell-1} \cup \Gamma_\ell,$$

which is the union of disjoint subset $\Gamma_i \subseteq \overline{\chi},\ i = 1,\ldots,\ell$ where the cardinality $|\Gamma_i| = K$ for $1 \leqslant i \leqslant \ell - 1$ and the cardinality of Γ_ℓ is smaller than K. In fact, ℓ is uniquely determined by the factorization of the integer $|\overline{\chi}| = (\ell-1)K + r$ where $0 \leqslant r < K$ is the remainder of the division $|\overline{\chi}|/K$. By the definition of χ, we see that $|\chi| \geqslant K$ and $|\chi| + |\overline{\chi}| = n_q$. Thus $|\overline{\chi}| \leqslant n_q - K$ and

$$\ell = 1 - \frac{r}{K} + \frac{|\overline{\chi}|}{K} \leqslant 1 + \frac{n_q - K}{K} = \frac{n_q}{K} \leqslant \frac{n - m(p-1)}{K} = \ell^*, \qquad (4.16)$$

where the last inequality follows from $n_q = n - \sum_{j\neq q} n_j < n - m(p-1)$ due to $m \ll n_j$ for all $j = 1,\ldots,p$. By using lemma 2.4, for every $1 \leqslant i \leqslant \ell$, one has

$$|u_\chi^T (A_q)_\chi^T (A_q)_{\Gamma_i} u_{\Gamma_i}| \leqslant \delta_{|\chi|+|\Gamma_i|} \|u_\chi\|_2 \|u_{\Gamma_i}\|_2.$$

Therefore,

$$|u_\chi^T (A_q)_\chi^T (A_q)_{\overline{\chi}} u_{\overline{\chi}}| = |\sum_{i=1}^{\ell} u_\chi^T (A_q)_\chi^T (A_q)_{\Gamma_i} u_{\Gamma_i}| \leqslant \sum_{i=1}^{\ell} \delta_{|\chi|+|\Gamma_i|} \|u_\chi\|_2 \|u_{\Gamma_i}\|_2 \leqslant \ell^* \delta_{|\chi|+K} \|u_\chi\|_2^2, \qquad (4.17)$$

where the last inequality follows from the fact $\delta_{|\chi|+|\Gamma_i|} \leqslant \delta_{|\chi|+K}$, $\ell \leqslant \ell^*$ and

$$\|u_{\Gamma_i}\|_2 \leqslant \|u_{\mathcal{L}_K(u)}\|_2 \leqslant \|u_\chi\|_2,$$

which follows from the fact $\mathcal{L}_K(u) \subseteq \chi$. □

The main result of GDP is summarized as follows.

Theorem 4.5. *Let $A = [A_1,\ldots,A_p] \in \mathbb{R}^{m\times n}$, where each $A_i \in \mathbb{R}^{m\times n_i}$ is a full-row-rank matrix with $m \ll n_i$. Let $x^* \in \mathbb{R}^n$ be a K-sparse vector, and let $y := Ax^* + \nu$ be the given observations with noise $\nu \in \mathbb{R}^m$. Suppose that the RIC of order $4K$ satisfies that*

$$\delta_{4K} < \frac{1}{2 + \ell^* + \omega(p-1)\left(1 + \omega^2\sqrt{p}\right)}, \qquad (4.18)$$

where $\ell^* = (n - m(p-1))/K > 0$ *and* $\omega = (\sqrt{5}+1)/2$. *((4.18) implies* $4K \leqslant m$.*) Then the sequence* $\{x^{(k)}\}_{k\geqslant 1}$*, generated by the GDP with an inner solver obeying assumption 4.1, satisfies that*

$$\|x^{(k+1)} - x^*\|_2 \leqslant \eta \|x^{(k)} - x^*\|_{(p,\infty)} + C_1(\|\nu\|_2 + \epsilon) + C_2\|A^T\nu\|_2, \tag{4.19}$$

where η, C_1 *and* C_2 *are positive constants given as*

$$\eta = \frac{\omega^3\sqrt{p}\,(p-1)\,\delta_{4K}}{\vartheta\,(1-\delta_{4K}^2)}, C_1 = \frac{\omega^3\sqrt{p\,(1+\delta_{4K})}}{\vartheta\,(1-\delta_{4K}^2)}, C_2 = \frac{1}{1-\delta_{4K}}\left(\frac{\omega}{\sqrt{1-\delta_{4K}^2}}+1\right), \tag{4.20}$$

where $\vartheta := 1 - (1 + \ell^* + (p-1)\omega)\delta_{4K} > 0$. *Condition (4.18) guarantees that* $\eta < 1$. *Moreover, if* $\nu = 0$ *and* $\epsilon = 0$*, then the sequence* $\{x^{(k)}\}_{k\geqslant 1}$ *converges to* x^*.

Proof: Let $\{x^{(k)}\}$ be the sequence generated by GDP with an inner solver satisfying assumption 4.1, where each $x^{(k)} = (x_1^{(k)}, \ldots, x_p^{(k)})$ with $x_j^{(k)} \in \mathbb{R}^{n_j}$ for $j = 1, \ldots, p$. As defined in GDP, $z_j^{(k)}$ is the solution produced by SPARSITY-SOLVER$(\phi_j^{(k)}, A_j)$, where $\phi_j^{(k)}$ is the vector defined at Step S2 of the algorithm. By assumption 4.1, we have $\|\phi_j^{(k)} - A_j z_j^{(k)}\|_2 \leqslant \epsilon$ and $|\mathrm{supp}(z_j^{(k)})| \leqslant m$. Define $\nu_j^{(k)} := \phi_j^{(k)} - A_j z_j^{(k)}$, which satisfies $\|\nu_j^{(k)}\|_2 \leqslant \epsilon$ for $j = 1, \ldots, p$. Let $\widetilde{x}^{(k)} = (\widetilde{x}_1^{(k)}, \ldots, \widetilde{x}_p^{(k)})$ be defined as in GDP. The first part of proof is to bound the term $\|\widetilde{x}^{(k)} - x^*\|_{(p,\infty)}$ in terms of $\|x^{(k)} - x^*\|_{(p,\infty)}$. By the definition 2.1, we see that

$$\|\widetilde{x}^{(k)} - x^*\|_{(p,\infty)} = \max_{1\leqslant j\leqslant p} \|\widetilde{x}_j^{(k)} - x_j^*\|_2 = \|\widetilde{x}_q^{(k)} - x_q^*\|_2$$

for some $q \in \{1, \ldots, p\}$. Notice that $y = Ax^* + \nu = \sum_{j=1}^p A_j x_j^* + \nu$, where $x_j^* \in \mathbb{R}^{n_j}$ for $j = 1, \ldots, p$. We also note that $\phi_q^{(k)} = y - \sum_{j<q} \widetilde{y}_j^{(k)} - \sum_{j>q} y_j^{(k)}$, where $\widetilde{y}_j^{(k)} = A_j \widetilde{x}_j^{(k)}$ and $y_j^{(k)} = A_j x_j^{(k)}$ by the structure of GDP. We have that

$$\begin{aligned} A_q\left(z_q^{(k)} - x_q^*\right) &= \phi_q^{(k)} - \nu_q^{(k)} - A_q x_q^* \\ &= \left(y - \sum_{j<q} \widetilde{y}_j^{(k)} - \sum_{j>q} y_j^{(k)}\right) - \nu_q^{(k)} - A_q x_q^* \\ &= \left(\sum_{j=1}^p A_j x_j^* + \nu - \sum_{j<q} \widetilde{y}_j^{(k)} - \sum_{j>q} y_j^{(k)}\right) - \nu_q^{(k)} - A_q x_q^* \\ &= \sum_{j<q} A_j\left(x_j^* - \widetilde{x}_j^{(k)}\right) + \sum_{j>q} A_j\left(x_j^* - x_j^{(k)}\right) + \nu - \nu_q^{(k)}. \end{aligned} \tag{4.21}$$

Define $S = \mathrm{supp}(x^*)$ and $S_j = \mathrm{supp}(x_j^*)$ for $j = 1, \ldots, p$. Clearly, $\sum_{j=1}^p |S_j| \leqslant K$ since $x^* = (x_1^*, \ldots, x_p^*)$ is K-sparse. Denote by $\Lambda_j^{(k)} := \mathcal{L}_K(z_j^{(k)} - x_j^*)$ and $\Omega_j^{(k)} = \mathcal{L}_K(z_j^{(k)})$. Clearly, $\mathrm{supp}(\widetilde{x}_j^{(k)}) = \mathrm{supp}(\mathcal{H}_K(z_j^{(k)})) \subseteq \Omega_j^{(k)}$. For the above-mentioned index q, we define the union

$$\chi := \Lambda_q^{(k)} \cup \Omega_q^{(k)} \cup S_q.$$

As $|\Lambda_q^{(k)}| = K$, $|\Omega_q^{(k)}| = K$ and $|S_q| \leqslant K$, we see that $|\chi| \leqslant 3K$. Let $\overline{\chi} = \{1, \ldots, n_q\} \setminus \chi$ be the complement of χ with respect to $\{1, \ldots, n_q\}$ which is the index set for the columns of A_q. Then we see that

$$A_q\left(z_q^{(k)} - x_q^*\right) = (A_q)_\chi\left(z_q^{(k)} - x_q^*\right)_\chi + (A_q)_{\overline{\chi}}\left(z_q^{(k)} - x_q^*\right)_{\overline{\chi}}. \tag{4.22}$$

Multiplying (4.21) by $\left[(A_q)_\chi (z_q^{(k)} - x_q^*)_\chi\right]^T$ leads to

$$\begin{aligned} (z_q^{(k)} - x_q^*)_\chi^T (A_q)_\chi^T A_q (z_q^{(k)} - x_q^*) &= \sum_{j<q} (z_q^{(k)} - x_q^*)_\chi^T (A_q)_\chi^T A_j (x_j^* - \widetilde{x}_j^{(k)}) \\ &\quad + \sum_{j>q} (z_q^{(k)} - x_q^*)_\chi^T (A_q)_\chi^T A_j (x_j^* - x_j^{(k)}) + (z_q^{(k)} - x_q^*)_\chi^T (A_q)_\chi^T (\nu - \nu_q^{(k)}). \end{aligned} \tag{4.23}$$

Substituting (4.22) into the left-hand side of (4.23) yields

$$\begin{aligned}
&\| \left(A_q\right)_\chi \left(z_q^{(k)} - x_q^*\right)_\chi \|_2^2 + \left(z_q^{(k)} - x_q^*\right)_\chi^T \left(A_q\right)_\chi^T \left(A_q\right)_{\overline{\chi}} \left(z_q^{(k)} - x_q^*\right)_{\overline{\chi}} \\
&= \sum_{j<q} \left(z_q^{(k)} - x_q^*\right)_\chi^T \left(A_q\right)_\chi^T A_j \left(x_j^* - \widetilde{x}_j^{(k)}\right) + \sum_{j>q} \left(z_q^{(k)} - x_q^*\right)_\chi^T \left(A_q\right)_\chi^T A_j \left(x_j^* - x_j^{(k)}\right) \\
&\quad + \left(z_q^{(k)} - x_q^*\right)_\chi^T \left(A_q\right)_\chi^T \left(\nu - \nu_q^{(k)}\right). \qquad (4.24)
\end{aligned}$$

We now estimate each term in (4.24). The first term on the left-hand side can be bounded from below by using the definition of RIC of order $|\chi| \leqslant 3K$:

$$\left(1 - \delta_{|\chi|}\right) \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2^2 \leqslant \| \left(A_q\right)_\chi \left(z_q^{(k)} - x_q^*\right)_\chi \|_2^2. \qquad (4.25)$$

We now bound the absolute value of the second term on the left-hand side of (4.24). Applying lemma 4.4 to $\chi = \Lambda_q^{(k)} \cup \Omega_q^{(k)} \cup S_q$ and $u = z_q^{(k)} - x_q^*$, we immediately obtain

$$| \left(z_q^{(k)} - x_q^*\right)_\chi^T \left(A_q\right)_\chi^T \left(A_q\right)_{\overline{\chi}} \left(z_q^{(k)} - x_q^*\right)_{\overline{\chi}} | \leqslant \ell^* \delta_{|\chi|+K} \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2^2, \qquad (4.26)$$

where $\ell^* = (n - m(p-1))/K$. Next, we bound every term on the right-hand side of (4.24). Note that for any $j \neq q$, one has

$$|\chi \cup S_j \cup \Omega_j^{(k)}| = |\Lambda_q^{(k)} \cup \Omega_q^{(k)} \cup S_q \cup S_j \cup \Omega_j^{(k)}| \leqslant 4K, \qquad (4.27)$$

since $|\Lambda_q^{(k)}| = K$, $|\Omega_q^{(k)}| = K$, $|\Omega_j^{(k)}| = K$, $S_q \cup S_j \subseteq S$ and $|S| \leqslant K$. As $\mathrm{supp}(x_j^* - \widetilde{x}_j^{(k)}) \subseteq S_j \cup \Omega_j^{(k)}$ and $j \neq q$, by using (4.27) and lemma 2.4, we have

$$\begin{aligned}
| \left[\left(z_q^{(k)} - x_q^*\right)_\chi\right]^T \left(A_q\right)_\chi^T A_j \left(x_j^* - \widetilde{x}_j^{(k)}\right) | &= | \left[\left(z_q^{(k)} - x_q^*\right)_\chi\right]^T \left(A_q\right)_\chi^T \left(A_j\right)_{S_j \cup \Omega_j^{(k)}} \left(x_j^* - \widetilde{x}_j^{(k)}\right)_{S_j \cup \Omega_j^{(k)}} | \\
&\leqslant \delta_{4K} \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2 \| \left(x_j^* - \widetilde{x}_j^{(k)}\right)_{S_j \cup \Omega_j^{(k)}} \|_2 \\
&\leqslant \omega \delta_{4K} \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2^2, \qquad (4.28)
\end{aligned}$$

where $\omega = (\sqrt{5}+1)/2$, and the last inequality follows from lemma 2.2, i.e.

$$\|x_j^* - \widetilde{x}_j^{(k)}\|_2 \leqslant \|x_q^* - \widetilde{x}_q^{(k)}\|_2 \leqslant \omega \| \left(x_q^* - z_q^{(k)}\right)_{\Omega_q^{(k)} \cup S_q} \|_2 \leqslant \omega \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2.$$

Let $V_j^{(k)} = \mathrm{supp}(x_j^{(k)})$. Since $x_j^{(k)}$ is K-sparse, by the same argument of (4.27), we have $|\chi \cup S_j \cup V_j^{(k)}| \leqslant 4K$. Thus for any $j \neq q$, one has

$$\begin{aligned}
&| \left[\left(z_q^{(k)} - x_q^*\right)_\chi\right]^T \left(A_q\right)_\chi^T A_j \left(x_j^* - x_j^{(k)}\right) | \\
&= | \left[\left(z_q^{(k)} - x_q^*\right)_\chi\right]^T \left(A_q\right)_\chi^T \left(A_j\right)_{S_j \cup V_j^{(k)}} \left(x_j^* - x_j^{(k)}\right)_{S_j \cup V_j^{(k)}} | \\
&\leqslant \delta_{4K} \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2 \| \left(x_j^* - x_j^{(k)}\right)_{S_j \cup V_j^{(k)}} \|_2 \\
&\leqslant \delta_{4K} \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2 \|x^* - x^{(k)}\|_{(p,\infty)}. \qquad (4.29)
\end{aligned}$$

Finally, the last term of the right-hand side of (4.24) is bounded as

$$\left[\left(z_q^{(k)} - x_q^*\right)_\chi\right]^T \left(A_q\right)_\chi^T \left(\nu - \nu_q^{(k)}\right) \leqslant \sqrt{1 + \delta_{|\chi|}} \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2 \|\nu - \nu_q^{(k)}\|_2. \qquad (4.30)$$

Substituting (4.25), (4.26), (4.28)–(4.30) into (4.24) yields

$$\left(1 - \delta_{|\chi|}\right) \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2^2 \leqslant \ell^* \delta_{|\chi|+K} \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2^2 + (q-1) \omega \delta_{4K} \| \left(z_q^{(k)} - x_q^*\right)_\chi \|_2^2$$

$$+(p-q)\delta_{4K}\|\left(z_q^{(k)}-x_q^*\right)_\chi\|_2\|x^*-x^{(k)}\|_{(p,\infty)}+\sqrt{1+\delta_{|\chi|}}\|\left(z_q^{(k)}-x_q^*\right)_\chi\|_2\|\nu-\nu_q^{(k)}\|_2.$$

Thus,

$$\begin{aligned}&\left[1-\delta_{|\chi|}-\ell^*\delta_{|\chi|+K}-(q-1)\omega\delta_{4K}\right]\|\left(z_q^{(k)}-x_q^*\right)_\chi\|_2\\&\leqslant(p-q)\delta_{4K}\|x^*-x^{(k)}\|_{(p,\infty)}+\sqrt{1+\delta_{|\chi|}}\|\nu-\nu_q^{(k)}\|_2.\end{aligned}\tag{4.31}$$

Notice that

$$1-\delta_{|\chi|}-\ell^*\delta_{|\chi|+K}-(q-1)\omega\delta_{4K}\geqslant\vartheta:=1-(1+\ell^*+(p-1)\omega)\delta_{4K},$$

which follows from $1\leqslant q\leqslant p$ and $\delta_{|\chi|}\leqslant\delta_{|\chi|+K}\leqslant\delta_{4K}$. It follows from (4.31) that

$$\vartheta\|\left(z_q^{(k)}-x_q^*\right)_\chi\|_2\leqslant(p-1)\delta_{4K}\|x^*-x^{(k)}\|_{(p,\infty)}+\sqrt{1+\delta_{4K}}\|\nu-\nu_q^{(k)}\|_2.$$

By applying lemma 2.2, we have

$$\begin{aligned}\|\widetilde{x}^{(k)}-x^*\|_{(p,\infty)}=\|\widetilde{x}_q^{(k)}-x_q^*\|_2=\|\mathcal{H}_K\left(z_q^{(k)}\right)-x_q^*\|_2&\leqslant\omega\|\left(z_q^{(k)}-x_q^*\right)_{\Omega_q^{(k)}\cup S_q}\|_2\\&\leqslant\omega\|\left(z_q^{(k)}-x_q^*\right)_\chi\|_2,\end{aligned}$$

where the last relation follows from $\Omega_q^{(k)}\cup S_q\subseteq\chi$. By noting that $\|\nu-\nu_q^{(k)}\|_2\leqslant\|\nu\|_2+\epsilon$, merging the above two inequalities yields

$$\|\widetilde{x}^{(k)}-x^*\|_{(p,\infty)}\leqslant\rho\|x^{(k)}-x^*\|_{(p,\infty)}+\frac{\omega\sqrt{1+\delta_{4K}}}{\vartheta}\left(\|\nu\|_2+\varepsilon\right),\tag{4.32}$$

where $\rho:=\frac{\omega(p-1)\delta_{4K}}{\vartheta}$. Let ρ_1,ρ_2 be given as in lemma 4.3 and η be given as in (4.20). Then

$$\omega^2\rho\rho_1\rho_2\sqrt{p}=\frac{1}{\sqrt{1-\delta_{3K}^2}}\cdot\frac{1}{\sqrt{1-\delta_{2K}^2}}\cdot\frac{\omega^3\sqrt{p}(p-1)\delta_{4K}}{\vartheta}\leqslant\eta,\tag{4.33}$$

where the inequality follows from the fact $\delta_{2K},\delta_{3K}\leqslant\delta_{4K}$. Merging (4.11), (4.32) and (4.33) yields

$$\begin{aligned}\|x^{(k+1)}-x^*\|_2\leqslant&\omega^2\rho\rho_1\rho_2\sqrt{p}\|x^{(k)}-x^*\|_{(p,\infty)}+\frac{\omega^3\rho_1\rho_2\sqrt{p(1+\delta_{4K})}\left(\|\nu\|_2+\epsilon\right)}{\vartheta}\\&+\left(\frac{\omega\rho_2}{1-\delta_{3K}}+\frac{1}{1-\delta_{2K}}\right)\|A^T\nu\|_2\leqslant\eta\|x^{(k)}-x^*\|_{(p,\infty)}+C_1\left(\|\nu\|_2+\epsilon\right)+C_2\|A^T\nu\|_2,\end{aligned}$$

where C_1 and C_2 are the constants given in (4.20). Thus the error bound (4.19) holds. It suffices to verify that $\eta<1$ under (4.18). As $\delta_{4K}<1$, we see that

$$\eta\leqslant\frac{\omega^3\sqrt{p}(p-1)\delta_{4K}}{(1-\delta_{4K})\left[1-(1+\ell^*+(p-1)\omega)\delta_{4K}\right]}.$$

We now verify that the right-hand side of the above is smaller than 1. This is equivalent to showing that

$$(1+\ell^*+(p-1)\omega)\delta_{4K}^2-\left[2+\ell^*+(p-1)\omega+\omega^3\sqrt{p}(p-1)\right]\delta_{4K}+1>0,$$

which is guaranteed if $\delta_{4K}<r^*$, where r^* is the smallest positive root to the equation in t:

$$(1+\ell^*+(p-1)\omega)t^2-\left[2+\ell^*+(p-1)\omega+\omega^3\sqrt{p}(p-1)\right]t+1=0.$$

Denote by $\zeta=2+\ell^*+\omega(p-1)(1+\omega^2\sqrt{p})$. It is easy to verify that

$$r^*=\frac{2}{\zeta+\sqrt{\zeta^2-4(1+\ell^*+(p-1)\omega)}}>\frac{1}{\zeta}=\frac{1}{2+\ell^*+\omega(p-1)\left(1+\omega^2\sqrt{p}\right)}.$$

Thus $\eta<1$ is guaranteed by (4.18). The proof is complete. □

In practice, the source signal x^* is generally not exactly K-sparse, but it is K-compressible (the signal has K significant components and the remaining ones are small but not necessarily zero). In this case, we are interested in reconstructing the best K-term approximation of x^*, i.e. $x_S^* = \mathcal{H}_K(x^*)$, where $S = \mathcal{L}_K(x^*)$. Since

$$y = Ax^* + \nu = Ax_S^* + \left(Ax_{\overline{S}}^* + \nu\right) = Ax_S^* + \overline{\nu},$$

where $\overline{\nu} = Ax_{\overline{S}}^* + \nu$, the observations y for a K-compressible signal x^* can be seen as those for the K-sparse signal x_S^* with noise $\overline{\nu}$. Applying theorem 4.5 to the system $y = Ax + \overline{\nu}$, we immediately obtain the following result.

Corollary 4.6. *Let $A = [A_1, \ldots, A_p] \in \mathbb{R}^{m\times n}$, where each $A_i \in \mathbb{R}^{m\times n_i}$ is a full-row-rank matrix with $m \ll n_i$ and $n_1 + \cdots + n_p = n$. Let $y := Ax^* + \nu$ be the given observations of $x^* \in \mathbb{R}^n$ with noise $\nu \in \mathbb{R}^m$. Suppose that the RIC of order 4K satisfies (4.18). Then the sequence $\{x^{(k)}\}_{k\geqslant 1}$, generated by the GDP algorithm with an inner solver obeying assumption 4.1, satisfies that*

$$\|x^{(k+1)} - x_S^*\|_2 \leqslant \eta\|x^{(k)} - x_S^*\|_{(p,\infty)} + C_1\left(\|\overline{\nu}\|_2 + \epsilon\right) + C_2\|A^T\overline{\nu}\|_2,$$

where $S = \mathcal{L}_K(x^*)$, $\overline{\nu} = Ax_{\overline{S}}^* + \nu$, and the constants C_1, C_2 and $\eta < 1$ are given in theorem 4.5. Moreover, if $\overline{\nu} = 0$ and $\epsilon = 0$, the sequence $\{x^{(k)}\}_{k\geqslant 1}$ converges to x_S^*.

Remark 4.7. The convergence analysis in this section is primarily of theoretical interest. It is well-known that the RIP is hard to verify, as shown in corollary 4 of [47]. However, when A is constructed randomly, conditions such as (4.18) can be guaranteed with high probability. For instance, if the entries of $A \in \mathbb{R}^{m\times n}$ are drawn independently from $\mathcal{N}(0, \frac{1}{m})$, then it follows from lemma 2 of [52] that all submatrices $A_i \in \mathbb{R}^{m\times n_i}$ specified in theorem 4.5 have full-row-rank almost surely; according to remark 9.28 of [21], the RIC of order $4K$ satisfies $\delta_{4K} \leqslant \delta_*$ with probability at least $1-\mu$ for a given $\delta_* \in (0,1)$, provided that

$$m \geqslant C\delta_*^{-2}\left\{4K\left[1 + \ln\left(\frac{n}{4K}\right)\right] + \ln\left(\frac{2}{\mu}\right)\right\},$$

where $\mu \in (0,1)$ is a given number and $C > 0$ is a universal constant. Thus, we mainly use Gaussian matrices in the numerical experiments demonstrated in the next section.

5. Numerical experiments

We evaluate the performance of the proposed algorithm using sparse signal reconstruction as an example of a LIP. The stopping criterion is set as $\|x^{(k)} - x^*\|_2/\|x^*\|_2 \leqslant 10^{-5}$, where $x^{(k)}$ is the solution generated by the algorithm and x^* is the target signal. When this criterion is met, the reconstruction is deemed a success. Given the flexibility in choosing inner solvers for GDP, we select one representative solver from each of the following three categories: convex optimization, heuristic methods, and thresholding methods. Specifically, we choose ℓ_1-minimization (ℓ_1-min), OMP, and HTP as representative inner solvers. Section 5.1.1 presents comparisons between ℓ_1-min and GDP with ℓ_1-min as inner solver, and between OMP and GDP with OMP as inner solver. Section 5.1.2 compares GDP (with HTP as inner solver) against several existing algorithms. Finally, we apply GDP to magnetic resonance imaging (MRI) reconstruction and compare its performance with several existing methods.

5.1. Experiments with random data

In our experiments, $A \in \mathbb{R}^{m\times n}$ is a Gaussian matrix with entries drawn independently from $\mathcal{N}(0, \frac{1}{m})$. When the noise vector $\nu \in \mathbb{R}^m$ is present, it is assumed to be a normalized Gaussian noise vector. The nonzero elements of the K-sparse signal $x^* \in \mathbb{R}^n$ are assumed to be drawn independently from $\mathcal{N}(0,1)$ and their positions follow the uniform distribution. The parameter $\tau = K$ is adopted in GDP for all experiments. For GDP_2, the ratio of n_1 to n_2 is set to be $1:4$, and for GDP_3, the ratio of n_1, n_2, and n_3 is set to be $1:1:3$.

5.1.1. GDP with ℓ_1-min and OMP as inner solvers

Throughout this section, ℓ_1-min is solved using CVX [25] with the *Mosek* solver [1]. We evaluate GDP performance with ℓ_1-min and OMP as inner solvers. The algorithm reconstructs a K-sparse vector $x^* \in \mathbb{R}^n$ from observations $y := Ax^* \in \mathbb{R}^m$ with $m = 1000$ and $n = 10000$. For each sparsity level $K = 100 + 5j$ where $j = 0, \ldots, 40$, we generate 50 random pairs (A, x^*) to estimate the success rate and calculate average runtime on successfully solved instances. In this experiment, GDP runs for up to 20 iterations on all test problems. The total number of iterations for OMP equals the sparsity level of x^*.

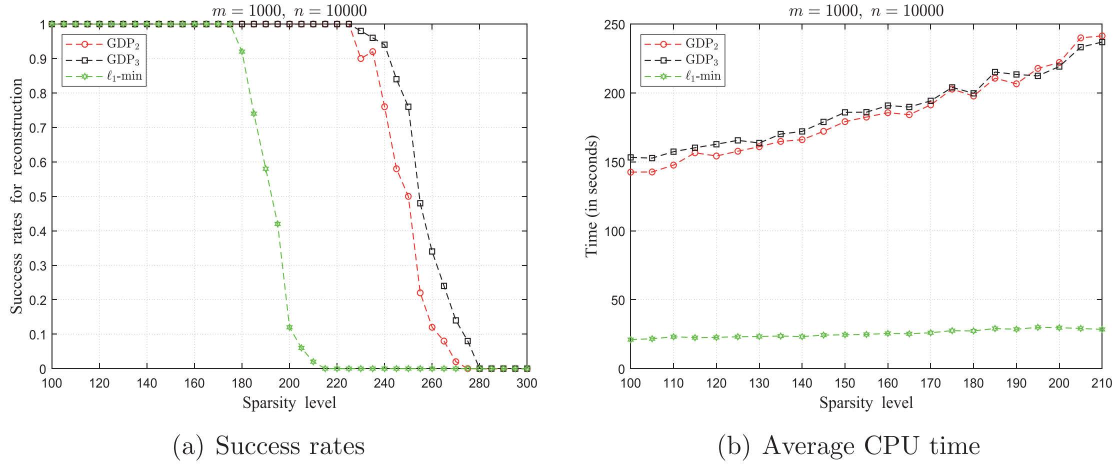


Figure 1. Comparison of success rates and average runtime (in seconds) of GDP_2, GDP_3 and ℓ_1-min for sparse reconstruction. GDP uses ℓ_1-min as inner solver.

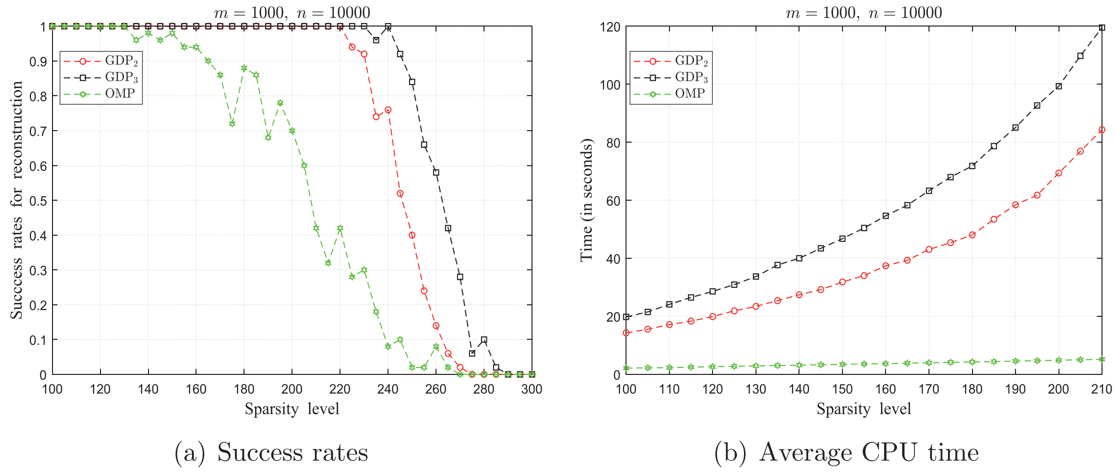


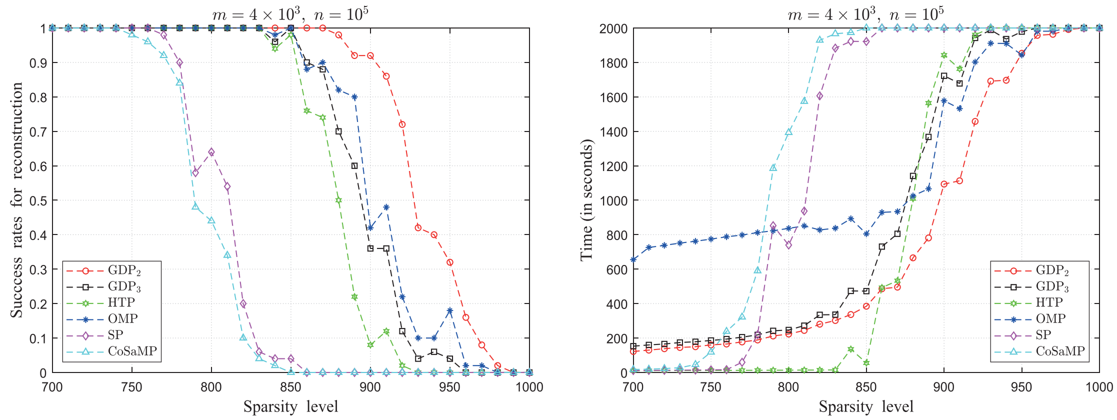
Figure 2. Comparison of success rates and average runtime (in seconds) of GDP_2, GDP_3 and OMP for sparse reconstruction. GDP uses OMP as inner solver.

The comparison between ℓ_1-min, GDP_2 and GDP_3 (the latter two using ℓ_1-min as inner solver) is illustrated in figure 1. This figure shows that GDP_2 and GDP_3 outperform ℓ_1-min in success rate for reconstructing sparse signals. Specifically, figure 1(a) indicates that ℓ_1-min can exactly reconstruct only signals with sparsity levels up to 175, while GDP_2 and GDP_3 can achieve exact reconstruction for signals with sparsity levels up to 225. However, figure 1(b) demonstrates that GDP_2 and GDP_3 spend more time than ℓ_1-min to complete reconstructions. Similarly, figure 2 compares OMP, GDP_2 and GDP_3 (the latter two using OMP as inner solver). Figure 2(a) shows that GDP_2 and GDP_3 achieve higher success rates than OMP, though they need more time to solve the problems, as shown in figure 2(b). These results demonstrate that the GDP framework, using either ℓ_1-min or OMP as inner solver, significantly enhances the likelihood of exact sparse signal reconstruction, enabling it to solve a much broader range of problems than traditional ℓ_1-min and OMP.

The experiment results also show that GDP exhibits no clear difference in problem-solving ability when using either ℓ_1-min or OMP as inner solver. However, the choice of inner solvers would directly affect computational cost for subproblems. Figures 1(b) and 2(b) indicate that GDP using ℓ_1-min as inner solver takes more time than using OMP as inner solver. Thus, it is advantageous to select an inner solver with low per-iteration complexity, such as the HTP which is a popular gradient-based thresholding method.

5.1.2. GDP with HTP as inner solver

In this section, we compare the performance of several mainstream algorithms with GDP (using HTP as inner solver) for sparse signal reconstruction from noisy observations. Specifically, the algorithms reconstruct a K-sparse vector $x^* \in \mathbb{R}^n$ from observations $y := Ax^* + \epsilon\nu \in \mathbb{R}^m$, where $m = 4 \times 10^3$, $n = 10^5$ and



(a) Success rates (b) Average CPU time

Figure 3. Comparison of success rates and average runtime (in seconds) of algorithms using inaccurate observations. GDP uses HTP as inner solver.

the noise level $\epsilon = 10^{-4}$. For each sparsity level $K = 700 + 10i$, $i = 0, 1, \ldots, 30$, we generate 50 random examples of (A, x^*, ν) to estimate the success rates of GDP$_2$, GDP$_3$, HTP, OMP, SP and CoSaMP. Each algorithm is limited to a maximum runtime of 2000 seconds. If an algorithm successfully recovers x^* within this limit, a *success* is recorded and its runtime is logged; otherwise, the runtime is recorded as 2000 s. As shown in section 4.2, the convergence of GDP does not require subproblems to be solved precisely at every iteration. Therefore, in our experiments, HTP is limited to at most 10 steps per global GDP iteration.

The results for success rates and average CPU time are shown in figure 3. This figure demonstrates that GDP$_2$ outperforms the other algorithms in success rate and takes less CPU time at high sparsity levels. Specifically, to achieve a 90% reconstruction success rate, SP and CoSaMP can only reconstruct vectors with sparsity level up to 770, while OMP, HTP and GDP$_3$ can reconstruct vectors with sparsity level up to 850. In contrast, GDP$_2$ is the most powerful method, capable of reconstructing vectors with sparsity level up to 900 while maintaining a success rate above 90%. The overall performance of GDP$_3$ is similar to that of OMP and HTP. Furthermore, figure 3(b) indicates that GDP$_2$ and GDP$_3$ also exhibit advantages in average CPU time, generally working faster than OMP. When the sparsity level is high, GDP$_2$ solves the problem faster than all other algorithms shown in the figure.

This experiment suggests GDP$_2$ outperforms GDP$_3$, though this conclusion should not be generalized due to limited problem sizes. The optimal number of blocks in matrix decomposition depends on subproblem size and available computational resources; large subproblems may require further decomposition to enable efficient solution by the inner solver.

5.2. Reconstruction of MRI image

In this subsection, we compare the performance of GDP$_2$, HTP, SP, and CoSaMP for reconstructing a 348 × 384 brain MRI image under a synthetic measurement model. The original T2-weighted Turbo Spin Echo axial image (figure 4(a)) is vectorized via column-wise stacking into $u^* \in \mathbb{R}^n$ with $n = 133{,}632$. This vector is transformed into a compressible representation $x^* \in \mathbb{R}^n$ through the orthogonal linear transform $\mathbf{W} = \mathbf{W}_2 \circ \mathbf{W}_1$, where $x^* = \mathbf{W}u^*$. Here, $\mathbf{W}_1$ denotes the discrete cosine transform and $\mathbf{W}_2$ denotes the discrete wavelet transform using a six-level 'sym8' wavelet with periodic extension. To reconstruct u^*, we acquire measurements $y := Au^* + \epsilon\nu \in \mathbb{R}^m$ with $m = n/4$ and $\epsilon = 0.5$, where A and ν are defined in section 5.1. Since $u^* = \mathbf{W}^{-1}x^*$, these measurements can be expressed equivalently as $y = A\mathbf{W}^{-1}x^* + \epsilon\nu$, enabling reconstruction in the transform domain. Let $\bar{x}$ denote the best K-term approximation of x^*, with sparsity level $K = \lceil 2m/7 \rceil$. Applying the reconstruction algorithm to the problem data $(y, A\mathbf{W}^{-1})$ recovers $\bar{x}$, from which the reconstructed image is obtained as $\bar{u} = \mathbf{W}^{-1}\bar{x}$.

We use the peak signal-to-noise ratio (PSNR) to evaluate the quality of reconstructed images. The PSNR is defined as $\mathrm{PSNR} := 10 \cdot \log_{10}(255^2/\mathrm{MSE})$, where MSE represents the mean-squared error

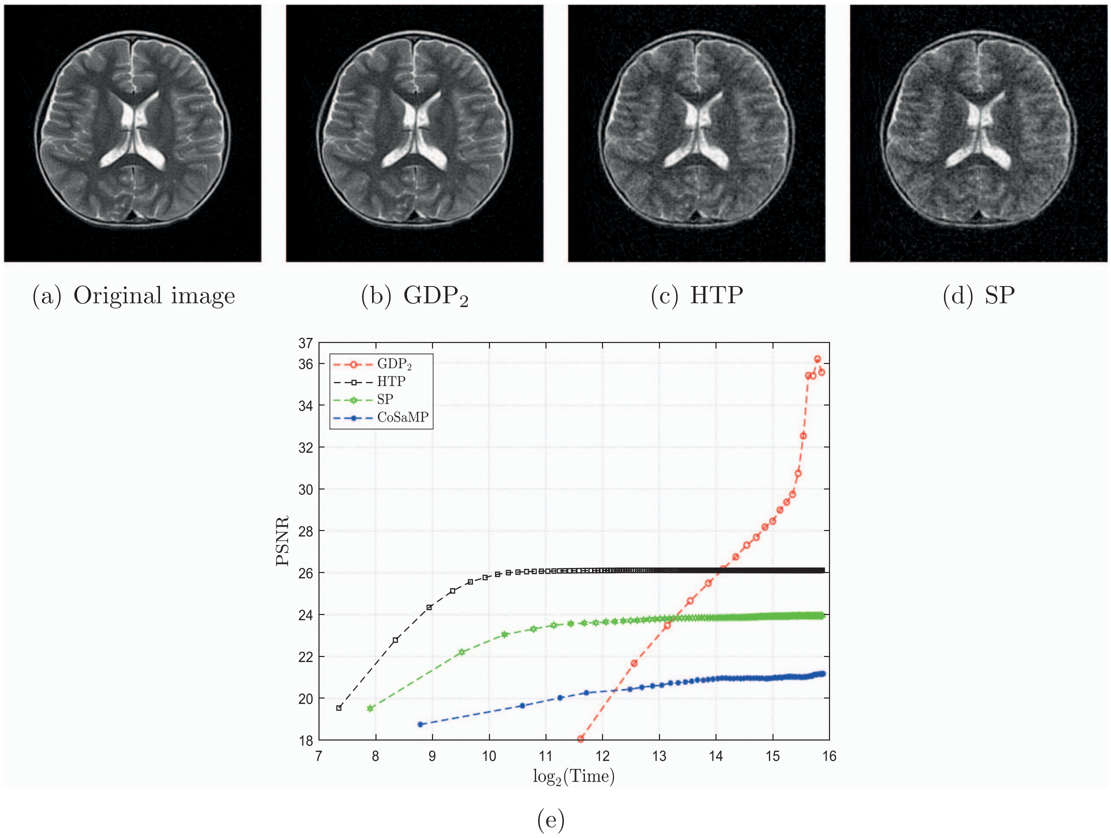


Figure 4. (a)–(d): Comparison of reconstructed images via GDP$_2$, HTP and SP. (Original image courtesy of the Shandong Zibo Central Hospital.) (e): PSNR (in dB) versus the logarithmic scale for CPU time in seconds.

between the true value u^* and its estimate $\bar{u}$. In this experiment, we applied GDP$_2$, HTP, CoSaMP and SP to reconstruct the MRI image. For GDP$_2$, the ratio of n_1 and n_2 was set to $2:1$, and its inner solver HTP was executed for only ten iterations. The maximum runtime allowed for all algorithms was set to 6×10^4 s. We monitored PSNR improvement during the iterations. The results are summarized in figure 4(e). The highest PSNR values achieved by GDP$_2$, HTP, SP and CoSaMP were approximately 36.2, 26.1, 24 and 21.2, respectively. Notably, only GDP$_2$ achieved a PSNR in the range $(26.1, 36.2]$. This clearly demonstrates that GDP$_2$ outperforms the other three algorithms (HTP, CoSaMP and SP) in this reconstruction task. The image reconstructed by GDP$_2$ was shown in figure 4(b). For comparison, the images generated by HTP and SP are displayed in figures 4(c) and (d), respectively. The image reconstructed by CoSaMP was omitted due to its low reconstruction quality. As shown in figures 4(a)–(d), GDP$_2$ achieves better visual quality in reconstructed images than the other algorithms. Note that this MRI experiment is for illustrative purposes only, and the proposed algorithm is not intended to surpass state-of-the-art algorithms for solving MRI tasks.

6. Conclusions

The GDP algorithm has been developed to find the sparse solutions of LIPs. It leverages those algorithms designed for small-scale problems to effectively solve large-scale problems. The GDP is shown to be globally convergent under the RIP. Numerical experiments have demonstrated that with an appropriate choice of inner solvers, GDP can outperform many existing sparsity-aware methods by achieving remarkably higher success rates for finding the true sparse solutions to LIPs. GDP is the first of its kind specifically developed to tackle large-scale problems that are challenging for the current sparsity-aware methods.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2023YFA1009300), National Natural Science Foundation of China (12471295, 12426306), Guangdong Basic and Applied Basic Research Foundation (2024A1515011566), the Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project (HZQSWS-KCCYB-2024016), and Shandong Provincial Natural Science Foundation (ZR2025MS99).

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/zhongfengsun/zhongfengsun.github.io/tree/gh-pages/data/GDP_IP [59].

Author contributions

Yun-Bin Zhao 0000-0002-2388-9047
Conceptualization (lead), Data curation (lead), Formal analysis (lead), Funding acquisition (lead), Investigation (lead), Methodology (lead), Project administration (lead), Resources (lead), Software (supporting), Supervision (lead), Validation (lead), Visualization (lead), Writing – original draft (lead), Writing – review & editing (lead)

Zhong-Feng Sun 0000-0003-1794-7869
Conceptualization (supporting), Data curation (supporting), Formal analysis (supporting), Funding acquisition (supporting), Investigation (supporting), Methodology (supporting), Software (lead), Validation (supporting), Visualization (supporting), Writing – original draft (supporting), Writing – review & editing (supporting)

Xin Liu 0000-0002-5705-0805
Funding acquisition (lead), Investigation (supporting), Methodology (supporting), Project administration (lead), Resources (lead), Software (supporting), Supervision (supporting), Writing – original draft (supporting), Writing – review & editing (supporting)

Quan Yu 0000-0002-8051-7477
Formal analysis (supporting), Investigation (supporting), Methodology (supporting), Software (supporting), Validation (supporting), Visualization (supporting)

References

[1] Andersen E D and Andersen K D 2000 The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm *High Performance Optimization* vol 33 (Springer) pp 197–232
[2] Al-Tawaha A, Cibaku E, Park S, Lavaei J and Jin M 2024 Distributed optimization and learning: a paradigm shift for power systems *Technical Report* Corpus ID: 270880680
[3] Bai Y, Chen W, Chen J and Guo W 2020 Deep learning methods for solving linear inverse problems: research directions and paradigms *Signal Process.* **177** 107729
[4] Beck A and Teboulle M 2009 A fast iterative shrinkage-thresholding algorithm for linear inverse problems *SIAM J. Imaging Sci.* **2** 183–202
[5] Bertsimas D and Johnson N 2024 Compressed sensing: a discrete optimization approach *Mach. Learn.* **113** 6725–64
[6] Blumensath T and Davies M E 2008 Iterative thresholding for sparse approximations *J. Fourier Anal. Appl.* **14** 629–54
[7] Böck B, Syed S and Utschick W 2024 Sparse Bayesian generative modeling for compressive sensing (arXiv:2411.09483v1)
[8] Boyd S, Parikh N, Chu E, Peleato B and Eckstein J 2010 Distributed optimization and statistical learning via the alternative direction method of multipliers *Found. Trends Mach. Learn.* **3** 1–122
[9] Bruckstein A M, Donoho D L and Elad M 2009 From sparse solutions of systems of equations to sparse modeling of signals and images *SIAM Rev.* **51** 34–81
[10] Bruckstein A M, Donoho D L and Elad M 2005 Decoding by linear programming *IEEE Trans. Inf. Theory* **51** 4203–15
[11] Candès E J, Wakin M B and Boyd S 2008 Enhancing sparsity by reweighted ℓ_1-minimization *J. Fourier Anal. Appl.* **14** 877–905
[12] Candès E J, Wakin M B and Boyd S 2007 The Dantzig selector: statistical estimation when p is much larger than n *Ann. Stat.* **35** 2313–51
[13] Chen S S, Donoho D L and Saunders M A 1998 Atomic decomposition by basis pursuit *SIAM J. Sci. Comput.* **20** 33–61
[14] Chen X, Liu J and Yin W 2024 Learning to optimize: a tutorial for continuous and mixed-integer optimization *Sci. China Math.* **67** 1191–262
[15] Chung J and Gazzola S 2024 Computational methods for large-scale inverse problems: a survey on hybrid projection methods *SIAM Rev.* **66** 205–84
[16] Dai W and Milenkovic O 2009 Subspace pursuit for compressive sensing signal reconstruction *IEEE Trans. Inf. Theory* **55** 2230–49

[17] Donoho D L and Huo X 2001 Uncertainty principles and ideal atomic decomposition *IEEE Trans. Inf. Theory* **47** 2845–62
[18] Donoho D L, Tsaig Y, Drori I and Starck J-L 2012 Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit *IEEE Trans. Inf. Theory* **58** 1094–121
[19] Elad M 2010 *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing* (Springer)
[20] Foucart S 2011 Hard thresholding pursuit: an algorithm for compressive sensing *SIAM J. Numer. Anal.* **49** 2543–63
[21] Foucart S and Rauhut H 2013 *A Mathematical Introduction to Compressive Sensing* (Springer)
[22] Foucart S and Subramanian S 2019 Iterative hard thresholding for low-rank recovery from rank-one projections *Linear Algebra Appl.* **572** 117–34
[23] Glowinski R 2014 On alternating direction methods of multipliers: a historical perspective *Modeling, Simulation and Optimization for Science and Technology* ed W Fitzgibbon, Y Kuznetsov, P Neittaanmäki and O Pironneau (Springer) pp 59–82
[24] Glowinski R, Osher S J and Yin W 2016 *Splitting Methods in Communuication, Imaging, Science and Engineering* (Springer)
[25] Grant M and Boyd S 2020 CVX: MATLAB software for disciplined convex programming (Version 2.2) (https://cvxr.com/cvx)
[26] Gohain P B and Jansson M 2023 Robust information criterion for model selection in sparse high-dimensional linear regression models *IEEE Trans. Signal Process.* **71** 2251–66
[27] Groetsch C 2011 Linear inverse problems *Handbook of Mathematical Methods in Imaging* ed O Scherzer (Springer)
[28] Hastie T, Tibshirani R and Wainwright M 2015 *Statistical Learning With Sparity: The Lasso and Generations* (CRC Press)
[29] Kumar K, Kamath A and Seelamantula C 2024 Tight-frame-like analysis-sparse recovery using nontight sensing matrices *SIAM J. Imaging Sci.* **17** 1587–618
[30] Lai M-J and Wang Y 2021 *Sparse Solutions of Underdetermined Linear Systems and Their Applications* (SIAM)
[31] Leinonen M, Codreanu M and Giannakis G B 2019 Compressed sensing with applications in wireless networks *Found. Trends Signal Process.* **13** 1–282
[32] Li X, Xie L and Hong Y 2022 Distributed aggregative optimization over multi-agent networks *IEEE Trans. Autom. Contr.* **67** 3165–71
[33] Li Z and Ding Z 2024 *Distributed Optimization and Learning: A Control-Theoretic Perspective* (Academic)
[34] Lin Z, Li H and Fang C 2022 *Alternating Direction Method of Multipliers for Machine Learning* (Springer)
[35] Luiken N and Leeuwen T V 2021 Relaxed regularization for linear inverse problems *SIAM J. Sci. Comput.* **43** S269–92
[36] Majumdar A 2016 *Compressed Sensing for Magnetic Resonance Image Reconstruction* (Cambridge University Press)
[37] Mallat S G and Zhang Z 1993 Matching pursuits with time-frequency dictionaries *IEEE Trans. Signal Process.* **41** 3397–415
[38] Needell D and Tropp J A 2009 CoSaMP: iterative signal recovery from incomplete and inaccurate samples *Appl. Comput. Harmon. Anal.* **26** 301–21
[39] Meng N and Zhao Y-B 2020 Newton-step-based hard thresholding algorithms for sparse signal recovery *IEEE Trans. Signal Process.* **68** 6594–606
[40] Pati Y C, Rezaiifar R and Krishnaprasad P S 1993 Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition *Proc. 27th Annu. Asilomar Conf. Signal, Systems and Computers* pp 40–44
[41] Richter M 2020 *Inverse Problems: Basics, Theory and Applications in Geophysics* (Birkhäuser)
[42] Starck J-L, Murtagh F and Fadili J M 2010 *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity* (Cambridge University Press)
[43] Sun Z-F, Zhao Y-B, Zhou J-C and Huang Z-H 2025 Dynamic thresholding algorithm with memory for linear inverse problems *Inverse Probl.* **41** 015004
[44] Sun Z-F, Zhou J-C and Zhao Y-B 2023 Heavy-ball-based optimal thresholding algorithms for sparse linear inverse problems *J. Sci. Comput.* **96** 93
[45] Temlyakov V 2000 Weak greedy algorithms *Adv. Comput. Math.* **12** 213–27
[46] Tillmann A N, Bienstock D, Lodi A and Schwartz A 2024 Cardinality minimization, constraints and regularization: a survey *SIAM Rev.* **66** 403–77
[47] Tillmann A-M and Pfetsch M-E 2014 The computational complexity of the restricted isometry property, the nullspace property and related concepts in compressed sensing *IEEE Trans. Inf. Theory* **60** 1248–59
[48] Tong C, Teng Y, Yao Y, Qi S, Li C and Zhang T 2021 Eigenvalue-free iterative shrinkage-thresholding algorithm for solving the linear inverse problems *Inverse Probl.* **37** 065013
[49] Tropp J A and Gilbert A C 2007 Signal recovery from random measurements via orthogonal matching pursuit *IEEE Trans. Inf. Theory* **53** 4655–66
[50] Wang J, Kwon S and Shim B 2012 Generalized orthogonal matching pursuit *IEEE Trans. Signal Process.* **60** 6202–16
[51] Wen J, He J, He Z and Liu X 2025 Heavy-ball enhanced pseudo-inverse-based hard thresholding algorithms for sparse linear inverse problems *Inverse Probl.* **41** 075003
[52] Yi J, Gao J, Wang T, Wu X and Xu W 2025 Outlier detection using generative models with theoretical performance guarantees *IEEE Trans. Inf. Theory* **71** 4012–31
[53] Zhao Y-B 2018 *Sparse Optimization Theory and Methods* (CRC Press)
[54] Zhao Y-B 2020 Optimal k-thresholding algorithms for sparse optimization problems *SIAM J. Optim.* **30** 31–55
[55] Zhao Y-B and Kocvara M 2015 A new computational method for the sparsest solutions to systems of linear equations *SIAM J. Optim.* **25** 1110–34
[56] Zhao Y-B and Li D 2012 Reweighted ℓ_1-minimization for sparse solutions to underdetermined linear systems *SIAM J. Optim.* **22** 1065–88
[57] Zhao Y-B and Luo Z-Q 2023 Improved RIP-based bounds for guaranteed performance of two compressed sensing algorithms *Sci. China Math.* **66** 1123–40
[58] Zhao Y-B and Luo Z-Q 2021 Analysis of optimal thresholding algorithms for compressed sensing *Signal Process.* **187** 108148
[59] Available at: https://github.com/zhongfengsun/zhongfengsun.github.io/tree/gh-pages/data/GDP_IP